User's Manual for

JetProg
Universal 48-pindrive Programmer,
expandable up to 256.

LabProg+
Universal 48-pindrive Programmer

SmartProg
Universal 40-pindrive Programmer with ISP capability

PREPROM-02aLV
Universal Memory Programmer

T51prog
MCS51 Series and Atmel AVR Microcontrollers Programmer with ISP capability

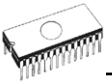
51&AVRprog
MCS51 Series and Atmel AVR Microcontrollers Programmer

PIKprog+
Microchip PICmicro Programmer with ISP capability

PIKprog
Microchip PICmicro Programmer

SEEprog
Serial EEPROM Programmer

ELNEC s.r.o.
Presov, Slovakia
January 2003



COPYRIGHT © 1997 - 2003
ELNEC s.r.o.

This document is copyrighted by ELNEC s.r.o., Presov, Slovakia. All rights reserved. This document or any part of it may not be copied, reproduced or translated in any form or in any way without the prior written permission of ELNEC s.r.o.

The control program is copyright ELNEC s.r.o., Presov, Slovakia. The control program or any part of it may not be analyzed, disassembled or modified in any form, on any medium, for any purpose.

Information provided in this manual is intended to be accurate at the moment of release, but we continuously improve all our products. Please consult manual on **www.elnec.com**.

ELNEC s.r.o. assumes no responsibility for misuse of this manual.

ELNEC s.r.o. reserves the right to make changes or improvements to the product described in this manual at any time without notice. This manual contains names of companies, software products, etc., which may be trademarks of their respective owners. ELNEC s.r.o. respects those trademarks.

ZLI-0017C

How to use this manual

Dear customer,

thank you for purchasing one of the ***ELNEC*** *programmer.*

This manual explains how to install the control program and how to use your programmer. It is assumed that the user has some experience with PCs and installation of software. Once you have installed the control program we recommend you consult the context sensitive HELP within the control program rather than the printed User's Manual. Revisions are implemented in the context sensitive help before the printed Users Manual.

Note: *Because this User's manual is common for all ELNEC programmers, read please section(s) respective programmer you have bought.*

This manual contains two main sections:

Quick Start

Read this section if you are an experienced user. You will find only specific information regarding installation of the control program and use of your programmer. For more detailed instructions you may read the **Description in detail** section or the **Troubleshooting** chapter for the respective programmer.

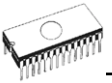
Descriptions in detail for each programmer and software

Read this section for the respective programmer if you are a less experienced user or if you need detailed information. You may find some less relevant features of programmer described here, but all programmer features are described in this section along with details regarding installation of the control program. Read this section to explore all of the features provided by your programmer.

Download please actual version of manual from ELNEC WEB site (www.elnec.com) if current one will be out of date.

Table of contents

How to use this manual ... 3
Introduction ... 9
Products configuration ... 13
PC requirements ... 14
Quick Start ... 15
JetProg ... 17
Introduction ... 18
JetProg elements ... 22
Connecting JetProg to the PC ... 23
Selftest and Calibration ... 25
Technical specification ... 26
LabProg+ ... 31
Introduction ... 32
LabProg+ elements ... 35
Connecting LabProg+ to the PC ... 36
Selftest and Calibration ... 38
Technical specification ... 39
SmartProg ... 43
Introduction ... 44
SmartProg elements ... 46
Connecting SmartProg to PC ... 47
In-system serial programming by SmartProg ... 48
Selftest and calibration ... 50
Technical specification ... 51
PREPROM-02aLV ... 55
Introduction ... 56
PREPROM-02aLV elements ... 58
Connecting PREPROM-02aLV programmer to PC ... 59
Selftest and calibration ... 61
PREPROM-02aLV specification ... 62
T51prog ... 65
Introduction ... 66
Connecting T51prog programmer to PC ... 68
In-System serial programming by T51prog ... 69
Selftest and calibration ... 71
T51prog specification ... 72
51&AVRprog ... 75
Introduction ... 76
Connecting 51&AVRprog programmer to PC ... 77
51&AVRprog specification ... 79
PIKprog+ ... 81
Introduction ... 82
Connecting PIKprog+ programmer to PC ... 84
In-System serial programming by PIKprog+ ... 85
Selftest and calibration ... 87
PIKprog+ specification ... 88
PIKprog ... 91
Introduction ... 92
Connecting PIKprog programmer to PC ... 93



PIKprog specification 95
SEEprog 97
Introduction 98
Connecting SEEprog programmer to PC 99
SEEprog specifications 101
Software 103
The programmer software 104
File 108
Device 112
Buffer 124
Options 130
Diagnostics 137
Help 138
Common notes 141
Software 142
Hardware 143
ISP (In-System Programming) 144
Other 148
Troubleshooting and warranty 149
Troubleshooting 150
If you have an unsupported target device 152
Warranty terms 153
Appendix 155
Appendix A - Device Problem Report form 157
Appendix B - Keep-Current Service 159
Appendix C - AlgOR service 161
Appendix D - registration card 163
Appendix E - CE certificates 165

Conventions used in the manual

References to the control program functions are in bold, e.g. **Load**, **File**, **Device**, etc. References to control keys are written in brackets **<>**, e.g. **<F1>.**

Terminology used in the manual:

Device any kind of programmable integrated circuits or programmable devices

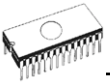
ZIF socket Zero Insertion Force socket used for insertion of target device

Buffer part of memory or disk, used for temporary data storage

Printer port type of port of PC (parallel), which is primarily dedicated for printer connection.

HEX data format - format of data file, which may be read with standard text viewers; e.g. byte 5AH is stored as characters '5' and 'A', which mean bytes 35H and 41H. One line of this HEX file (one record) contains start address and data bytes. All records are secured with checksum.

Introduction



This user's manual covers all ELNEC programmers: **JetProg, LabProg+, SmartProg, PREPROM-02aLV, T51prog, 51&AVRprog, PIKprog+, PIKprog** and **SEEprog**.

JetProg is a universal programmer and logic IC tester with 48 powerful pindrivers in base configuration, expandable up to 256. This design allows easily add new devices to the device list. Provides very competitive price but excellent hardware design for reliable programming. Best "value for money" in this class.

LabProg+ is a universal programmer and logic IC tester with 48 powerful pindrivers. This design allows easily add new devices to the device list. LabProg+ is a true universal and a true low cost programmer, providing one of the best "value for money" in today's market.

SmartProg is a small, fast and powerful programmer of all kinds of programmable devices. Using build-in in-circuit serial programming (ISP) connector the programmer is able to program ISP capable chips in-circuit. It has design, which allows easily add new devices to the device list. Nice "value for money" in this class.

PREPROM-02aLV is a small and powerful EPROM, EEPROM, Flash EPROM and serial EEPROM programmer and static RAM tester, designed for professional mobile applications. In addition, PREPROM-02aLV programmer with auxiliary modules support also microprocessors (MCS48, MCS51, PIC, AVR), GALs, etc. Programmer can work with 'true LV' device too - from 2V.

T51prog is little, powerful and very fast portable programmer for MCS51 series and Atmel AVR microcontrollers with ISP capability. T51prog enables also programming serial EEPROM with interface types IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx).

51&AVRprog is little and portable programmer for MCS51 series and Atmel AVR microcontrollers. 51&AVRprog enables also programming serial EEPROM with interface types IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx).

PIKprog+ is little, very fast and powerful portable programmer for PICmicro® family microcontrollers and serial EEPROM with IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx) interface types. Using build-in in-circuit serial programming (ISP) connector programmer is able to program PICmicro® family microcontrollers using serial algorithms.

PIKprog is little and portable programmer for Microchip PICmicro series of microcontrollers. PIKprog enables also programming serial EEPROM with interface types IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx).

SEEprog is universal programmer of all serial EEPROM in 8 pin DIL package. SEEprog programs EEPROM with interface IIC, SPI and Microwire, and also specialty as for example digital thermometers. The programmer supports LV (3.3V) devices too.

All programmers of our works with almost any IBM compatible PC, AT or higher, portable or desktop personal computers. No special interface card is required to connect to the PC, since programmers use the parallel (printer) port.

All programmers function flawlessly on systems running DOS, Windows 3.x, Windows 95/98/Me/NT/2000/XP.

All programmers are driven by an **easy-to-use, control program** with pull-down menus, hot keys and online help. There are available two SW versions - for DOS (PG4U) and for WIN95/98/Me/NT/2000/XP (PG4UW). Control program is common for all the ELNEC's programmers (JetProg, LabProg+, SmartProg, PREPROM-02aLV, T51prog, 51&AVRprog, PIKprog+, PIKprog and SEEprog).

Advanced design, including protection circuits, original brand components and careful manufacturing allows us to provide a **one-year** (T51prog, 51&AVRprog, PIKprog+, PIKprog and SEEprog) to **three-year** (JetProg, LabProg+, SmartProg, PREPROM-02aLV) **warranty** on parts and labor for the programmers (limited 25,000 cycle warranty on ZIF socket).

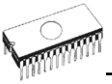
Free additional services:

- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.

Free software updates are available from our Internet address **www.elnec.com.**

We also offer the following new services in our customer support program: Keep-Current and AlgOR.

- **Keep-Current** is a service by which ELNEC ships to you the latest version of the control program for programmer and the updated user documentation. A Keep-Current service is your



hassle-free guarantee that you always have access to the latest software and documentation, at minimal cost.

- **AlgOR** (Algorithm On Request) service allows you to receive from ELNEC software support for programming devices not yet available in the current device list.

Note: *We don't recommend use programmers LabProg+, PREPROM-02aLV, 51&AVRprog, PIKprog and SEEprog for In-circuit programming. See FAQ on site* **www.elnec.com**.

Products configuration

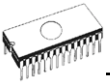
Before installing and using your programmer, please carefully check that your package includes all next mentioned parts. If you find any discrepancy with respective parts list and/or if any of these items are damaged, please contact your distributor immediately.

JetProg, LabProg+, SmartProg, PREPROM-02aLV, T51prog and PIKprog+ programmer configuration

- programmer
- cable with two 25 pin, D-type connectors for connecting the programmer to the PC
- external power supply (suitable for respective programmer)
- diagnostic POD for selftest of programmer
- cap for ZIF socket (anti-dust cover)
- this User's manual
- CD with the control program and additional files
- "DEVICE PROBLEM REPORT" form
- registration card
- shipping case

51&AVRprog, PIKprog and SEEprog programmer configuration

- programmer
- cable with two D-type connectors for connecting the programmer to the PC
- external power supply (suitable for respective programmer)
- this User's manual
- CD with the control program and additional files
- "DEVICE PROBLEM REPORT" form
- registration card
- shipping case



PC requirements

Minimal PC requirements

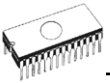
- PC AT 10 MHz (DOS version software), PC 486 (WIN version software)
- 512 KB free RAM (DOS version software), 16MB RAM (WIN version software)
- one CD drive
- HDD, 10 MB free space
- operating system MS/PC-DOS version 3.2 or later, WIN 3.xx, WIN95/98/Me/NT/2000/XP
- one free printer port with nothing attached

Recommended PC requirements

- Pentium PC 100MHz or higher
- 32 MB free RAM
- one CD drive
- HDD with minimum 10 MB free space
- operating system: MS Windows 95/98/Me/NT/2000/XP
- one free bi-directional printer port with nothing attached
- for JetProg, SmartProg, T51prog and PIKprog+ free parallel (printer) port on PCI bus, IEEE 1284 compatible (ECP/EPP)

Note: *For convenience, we suggest that you use a supplementary multi I/O card to provide an additional printer port (LPT2 for example), in order to avoid sharing the same LPT port between printer and programmer.*

Quick Start



Installing programmer hardware

- switch off the PC and programmer
- connect the communication port of programmer to a printer port of PC using cable supplied
- switch on the PC
- connect the connector of the power supply adapter to the programmer

Installing the programmer software

Run the installation program from the CD (DOS: Install.exe, WIN: Setup.exe) and follow the on-screen instructions. Please, for latest information about the programmer hardware and software see **www.elnec.com**.

Using programmer software

Launch PG4U.EXE (DOS) or PG4UW.EXE (WIN) to enter the control program. The menu **Device** contains the device manipulation commands. The menu **File** contains commands for files and directories. The menu **Buffer** is to be used for buffer manipulation.

Programming a device - the shortest way

Use the hot key **<Alt+F5>** to input the device name and/or manufacturer to select the desired type of target device. If you want to copy an existing device, insert it into the ZIF socket of the programmer and then press key **<F7>**. If you want to program a target device with data from a disk press key **<F3>** and read the appropriate file into the buffer. Then insert your target device into the ZIF socket. To check if the device is blank - press key **<F6>.** Now you can program the device by pressing key **<F9>.** After programming you may perform additional verification by pressing key **<F8>.**

JetProg

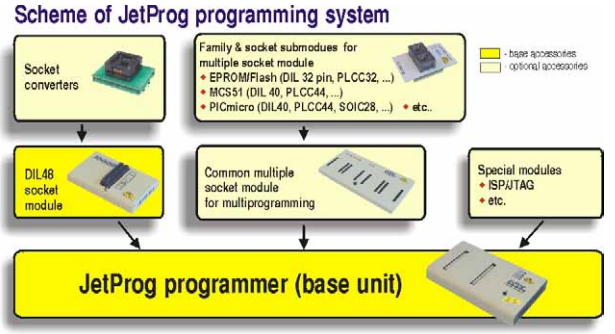
Introduction

JetProg is a new generation of Windows 95/98/Me/NT/2000/XP based ELNEC universal programmers built to meet the rigorous demands of the leading engineers and programming centers.

JetProg supports all kinds of types and silicon technologies of programmable devices. It provides very competitive price but excellent hardware design for reliable programming. Best "value for money" in this class.

JetProg interfaces with the IBM PC, AT or higher, portable or desktop personal computers. Programmer allows you to directly connect to your PC through **any standard parallel (printer) port** (no special interface card needed). We recommend to use **parallel (printer) port on PCI bus, IEEE 1284 compatible (ECP/EPP)**. The JetProg control program support standard IEEE1284 also.

JetProg offer very fast programming due high-speed FPGA driven hardware and support of ECP/EPP parallel port. Consequently and due special protocol is communication between PC and JetProg programmer fast and very reliable. The programming AT29C040A takes about 28 seconds it is faster than most its competitors. As a result, this programmer is optional solution for middle quantities programming in production or programming centers.



JetProg, base configuration

- JetProg, base unit
- JetProg, DIL48 socket module

For following text, term JetProg means JetProg in base configuration.

JetProg has 48 powerful pindrivers in base unit, expandable up to 256 pindrivers using "pindriver expansion" modules. Advanced pin drivers incorporate high-quality high-speed circuitry to deliver programming and testing performance without overshoot or ground bounce for all device technologies. Pin drivers operate down to 1.8V so you'll be ready to program the full range of today's advanced low-voltage devices.

Modular design of JetProg allows adapting the programmer according to customers needs either as very flexible universal programmer for laboratory or high efficient multiprogrammer in production line. Multiprogramming capability for most of supported devices is accomplished by using "multiple socket" modules.

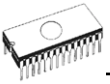
Powerful pindriver provides logic level, pull-up/pull-down, clock, ground, one VCC supply and two programming supply and, certainly read, on each of all 48 pins independently. This advanced design give it the ability to program almost every programmable device in DIL up to 48 pins without adapter or family specific module. Support for today and tomorrow programmable devices gives engineers the freedom to choose the optimum device for new design.

JetProg isn't only programmer, but also **tester** of TTL/CMOS logic ICs and memories. Furthermore, it allows generate of user-definable **test pattern sequences**.

The programmer has on-board intelligence, comprise of powerful microcontroller system and support devices. JetProg has been designed for **multitasking operating systems** and is able to perform time-critical programming sequences independently of the PC operating system status and without being interrupted by any another parallel process running on the PC. Consequently, JetProg works without any problem on systems running Windows 3.x, Windows 95/98/Me/NT/2000/XP.

The programmer performs **device insertion test** (wrong or backward position) and contact check (poor contact pin-to-socket) before it programs each device. These capabilities, supported by overcurrent protection and signature-byte check help prevent chip damage due to operator error.

Built-in protection circuits eliminate damage of programmed device due to mains supply fluctuations, communication error or if PC is frozen. In event of such errors microcontroller in programmer performs, independently on the PC, exactly specified sequence of steps, so that programmed target device remains intact. Programmer's hardware offers enough resources for **selftest**, that control program is any time be able



to check pindrivers, present and correct level of all voltages, check the timing and communication between programmer and PC.

An optimally designed printed circuit minimizes negative programming effects at the socket (such as ground bouncing, supply voltage instability). All the inputs of the JetProg programmer, including the ZIF socket, connection to PC and power supply input, are **protected against ESD** to protect the programmer and programmed circuits against damage due ESD.

JetProg performs programming **verification** at the marginal level of supply voltage, which, obviously, improves programming yield, and guarantees long data retention.

The JetProg programmer is driven by an **easy-to-use control program** with pull-down menu, hot keys and online help. You can select target device by its class, by manufacturer or simply by typing a fragment of vendor name and/or part number. Besides the standard device-related commands (read, blank check, program, verify, erase) have been implemented some test functions (insertion test, signature-byte check), and some special functions (autoincrement, production mode - start immediately after insertion of chip into socket). The control program permits data manipulation within the buffer and between the buffer and files and performs automatic file format detection and conversion. The software is available in English and in more another languages.

The rich-featured **auto-increment function** of JetProg enables you to assign individual serial numbers to each programmed device. This function simply increments a serial number in the buffer each time a new device is inserted. Furthermore, the function enables the operator to read serial numbers and/or any programmed device identification signatures from a file.

Various **socket converters** are available for the JetProg, DIL to PLCC, DIL to PSOP, DIL to SOIC, DIL to TQFP and DIL to TSOP. Software controlled pin-swapping capability together with universal converters are what give the JetProg the ability to program the target device in any package type. For example, just one DIL44 to PLCC44 converter will allow the JetProg to handle any PLCC44 target device in the current device list.
Devices with more than 48 pins are supported by

- pindriver expansion module and universal single socket module
- simple special package converters

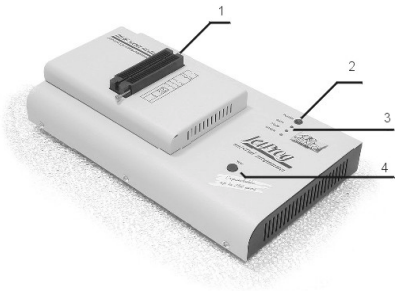
Note: S*imple adapter usually available from ELNEC or easily built by the user from information provided in the file "adapters.txt" included with the control program*.

It is important to remember that in most cases new devices require **only a software upgrade** since the JetProg has 48 true pin drivers, which can perform as required under program control. With our prompt service new devices can be added to the current list within hours!

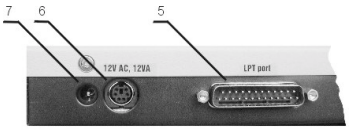
Advanced design including protection circuits, original brand components and careful manufacturing allows us to provide a **three-year warranty** on parts and labor for the JetProg (limited 25,000-cycle warranty on ZIF socket).

JetProg elements

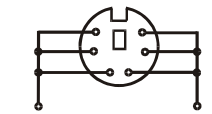
① DIL48 socket module with 48 pin ZIF socket
② LED indicator power/sleep
③ LED indicators for work result
④ YES! Button



⑤ Connector for PC <-> JetProg communication cable
⑥ Power supply connector
⑦ Internal use connector



Power supply connector



AC	~	~	12VAC/1A
DC	+	-	15VDC/1A
DC	-	+	15VDC/1A

Note: *When the JetProg programmer isn't active, control program switch the programmer into "sleep mode". In this state it consumes less than 1W of power. Therefore the programmer has no On/Off switch. When the power LED indicator glows with a low intensity the JetProg is in sleep mode.*

Connecting JetProg to the PC

Switch off PC and programmer. Insert the communication cable included with your JetProg programmer package to a free printer port on your PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counter-connectors. This is very important. It may be uncomfortable to switch between printer cable and programmer cable, though it is not recommended to operate the JetProg programmer through a mechanical printer switch. Use of an electronic printer switch is impossible. But you can install a second multi-I/O in your computer, thus obtaining a supplementary printer port, says LPT2. So your printer may remain on LPT1 while the programmer on LPT2.

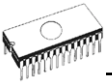
Switch on the PC.
Connect the mains connector of the power supply (or the wall-plug power supply itself) to a mains plug, and then connect the mini-DIN connector to the programmer's connector labeled "12VAC". At this time all 'work result' LEDs (and 'POWER' LED) light up successive and then switch off. Once the POWER LED lights with low brightness then the JetProg programmer is ready to run.
Next run the control program for JetProg.

Caution! *If you don't want to switch off your PC when connecting the JetProg, proceed as follows*:

- ***When connecting** the programmer to the PC: **FIRST** insert the **communications cable** and **THEN** the **power-supply connector.***
- ***When disconnecting** the programmer from the PC: **FIRST** disconnect the **power-supply connector** and **THEN** the **communication cable**.*

From JetProg's point of view the connecting and disconnecting sequence is irrelevant. Protection circuits on all programmer inputs keep it safe. **But think of your PC please**.

Note: *Programmer's protection electronics protect the target device and the programmer itself against either short or long-term power failures and, partly, also against a PC failure. However, it is not possible to grant the integrity of the target device due to incorrect, user-selected programming parameters. Target device may be not destroyed by forced interruption of the control program (reset or switch-off PC), by removing the physical connection to the programmer, but the content of actually programmed cell may remains undefined.*



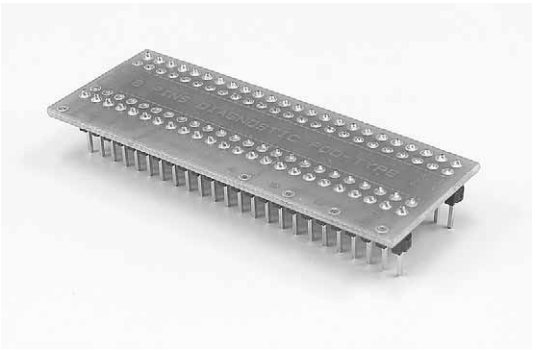
Don't unplug the target device from the ZIF socket during work with devices (LED BUSY shine).

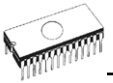
Problems related to the JetProg ⇔ PC interconnection, and their removing

If you have any problems with JetProg ⇔ PC interconnection, see section ***Common notes*** please.

Selftest and Calibration

If you feel that your programmer does not react according to your expectation, please run the programmer selftest using Diagnostic POD, enclosed with the standard delivery package. For optimal results with programmer we recommend you undertake every 6 months, an extended test and to check the calibration. See instructions for selftest in the **Diagnostics** menu of PG4UW.





Technical specification

HARDWARE

Base unit, DACs

- FPGA based IEEE 1284 slave printer port, up to 1MB/s transfer rate
- on-board powerful microprocessor (20MHz) supported by FPGA based state machine, 20MHz powered
- three D/A converters for VCCP, VPP1, and VPP2, controllable rise and fall time
- VCCP range 0..8V/1A
- VPP1, VPP2 range 0..26V/1A
- autocalibration
- selftest capability
- protection against surge and ESD on power supply input, parallel port connection
- banana jack for ESD wrist straps

Socket, pindriver

- pindrivers: 48 as standard, max. 256
- 1x VCC, 2x VPP can be connected to each pin
- perfect ground for each pin
- FPGA based TTL driver provides H, L, CLK, pull-up, pull-down on all pindriver pins
- analog pindriver output level selectable from 1.8 V up to 26V
- current limitation, overcurrent shutdown, power failure shutdown
- ESD protection on each pin of socket (IEC1000-4-2: 15kV air, 8kV contact)
- continuity test: each pin is tested before every programming operation

Socket, base configuration

- 48-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 48-pin

DEVICE SUPPORT

Programmer

- EPROM: NMOS/CMOS, 1702*, 2708*, 27xxx and 27Cxxx series, with 8/16 bit data width, full support for LV series
- EEPROM: NMOS/CMOS, 28xxx, 28Cxxx, 27EExxx series, with 8/16 bit data width

- Flash EPROM: 28Fxxx, 29Cxxx, 29Fxxx, 29BVxxx, 29LVxxx, 29Wxxx, 49Fxxx series, from 256Kbit to 32Mbit, with 8/16 bit data width, full support for LV series
- serial E(E)PROM: 17Xxxx, 18Vxxx, 24Cxxx, 24Fxxx, 25Cxxx, 45Dxxx, 59Cxxx, 85xxx, 93Cxxx, EPCxxx, NVM3060, MDAxxx series, full support for LV series
- PROM: Harris, National, Philips/Signetics, Tesla, TI
- NV RAM: Dallas DSxxx, SGS/Inmos MKxxx, SIMTEK STKxxx, XICOR 2xxx, ZMD U63x, , Benchmarq BQxxx, Catalyst (CSI) CAT22Cxx, Greenwitch Instr. GRxxx, Ramtron FMxxxx, Seiko Instruments S-22xxx, SGS-Thomson (STM) Mxxxxx series
- PLD: SPLD, CPLD series: Altera, Atmel, AMD-Vantis, Lattice, NS, Philips, TI, Xilinx...
- microcontrollers 48 series: 87x41, 87x42, 87x48, 87x49, 87x50 series
- microcontrollers 51 series: 87xx, 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx, all manufacturers, Philips 87C748...752, 87LPCxxx series
- microcontrollers Intel 196 series: 87C196 KB/KC/KD/KT/KR...
- microcontrollers Atmel AVR: AT90Sxxxx, ATtiny, ATmega series
- microcontrollers Microchip PICmicro: PIC12Cxxx, PIC12Fxxx, PIC16C5x, PIC16Cxxx, PIC16Fxxx, PIC17Cxxx, PIC18Cxxx series
- microcontrollers Motorola: 68HC05, 68HC08, 68HC11 series
- microcontrollers National: COP8xxx series
- microcontrollers SGS-Thomson: ST6xx series
- microcontrollers ZILOG: Z86xxx series
- microcontrollers others: Holtek, Samsung, Scenix, Fujitsu, Hitachi, NEC, Toshiba, Cypress, TI

Notes:

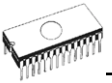
- *Devices marked * are obsolete, programming with additional module*
- *For all supported devices see actual **Device list***

I.C. Tester

- TTL type: 54,74 S/LS/ALS/H/HC/HCT series
- CMOS type: 4000, 4500 series
- static RAM: 6116.. 624000
- user definable test pattern generation

Package support

- package support includes DIP, PLCC, SDIP, SOIC, PSOP, TQFO, TSOP and other
- support all devices in DIP with default socket
- support devices in non-DIP packages up to 48 pins with universal adapters
- programmer is compatible with third-party adapters for non-DIP support



Programming speed

Note. *These times strongly depend on PC speed, LPT port type and operating system free resources. Therefore are given values of two different PC configurations for comparison.*

Device	Operation	Time A	Time B
27C010	programming and verify	24 sec	20 sec
AT29C040A	programming and verify	38 sec	28 sec
AM29F040	programming and verify	49 sec	42 sec
PIC16C67	programming and verify	8 sec	7 sec
PIC18F452	programming and verify	13 sec	9 sec

Time A conditions: *Pentium MMX, 250 MHz, ECP/EPP, WIN98.*
Time B conditions: *Athlon, 750 MHz, ECP/EPP on PCI bus, WIN98.*

SOFTWARE

- **Algorithms:** only manufacturer approved or certified algorithms are used. Custom algorithms are available at additional cost.
- **Algorithm updates:** software updates are available approx. every 2 weeks, free of charge.
- **Main features:** revision history, session logging, on-line help, device and algorithm information

Device operations

- **standard:**
  - intelligent device selection by device type, manufacturer or typed fragment of part name
  - automatic ID-based selection of EPROM/Flash EPROM
  - blank check, read, verify
  - program
  - erase
  - configuration and security bit program
  - illegal bit test
  - checksum
- **security**
  - insertion test, reverse insertion check
  - contact check
  - ID byte check
- **special**
  - production mode (automatic start immediately after device insertion)
  - auto device serial number increment
  - statistic
  - count-down mode

Buffer operations

- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

File load/save

- file load/save
- no download time because programmer is PC controlled
- automatic file type identification

Supported file formats

- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S-record, MOS, Exormax, Tektronix, ASCII-space-HEX
- Altera POF, JEDEC (ver. 3.0.A), e.g. from ABEL, CUPL, PALASM, TANGO PLD, OrCAD PLD, PLD Designer ISDATA, etc.

PC system requirements

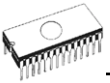
See section ***Introduction/ PC requirements***

GENERAL

- operating voltage 12..15V AC/max. 1A (adapter included in package)
- power consumption max. 12W active, about 1.5W sleep
- dimensions 275L x 157W x 47H mm (10,8"L x 6,2"W x 1,85"H)
- weight (without external adapter) 1.8kg
- temperature 5°..40°C
- humidity 20%..80%, non condensing

Base configuration package includes

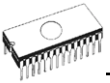
- JetProg, base unit
- JetProg, DIL48 socket module
- connection cable
- diagnostic POD for selftest
- anti-dust cover for ZIF socket
- switched power supply adapter 15V DC/1A
- user manual
- software
- “DEVICE PROBLEM REPORT" form
- registration card
- transport case



Additional services

- Keep Current.
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.

LabProg+



Introduction

LabProg+ is a universal **programmer** that supports programmable integrated circuits or devices manufactured in various technologies. Powerful internal pin-driver electronics controls logic levels, pull-up/pull-down, clock, ground, one power supply and two programming supplies and is able to read all 48 pins independently. This advanced design gives LabProg+ the ability to handle almost every programmable device in DIL package up to 48 pins without any adapters and/or family modules. This design philosophy allows ELNEC to easily add new devices to the device list, giving you the freedom to implement the optimum device in your designs.

LabProg+ is also a **tester** of TTL/CMOS logic circuits and various memories. Furthermore, it can generate user-definable **test pattern sequences**. LabProg+ is a true universal and a true low-cost programmer, providing the best "value for money" in today's market.

LabProg+ works with almost any IBM compatible PC, AT or higher, portable or desktop personal computers. No special interface card is required to connect to the PC since LabProg+ uses the **standard parallel printer port**. The LabProg+ control program also supports bi-directional protocols for the parallel connection to the PC printer port providing fast and reliable **communication speed**.

The programmer has on-board intelligence and is controlled by powerful microcontroller system and support devices. LabProg+ has been designed for **multitasking operating systems** and is able to perform time-critical programming sequences independently of the PC operating system status and without being interrupted by any other parallel process running on the PC. Consequently, LabProg+ functions flawlessly on systems running Windows 3.x, Windows 95/98/Me/NT/2000/XP.

LabProg+ performs **device insertion test** (wrong or backward position) and contact check (poor contact pin-to-socket) before it programs any device. These capabilities, supported by current limit protection and signature-byte check, help prevent chip damage due to operator error.

Built-in protection circuits help prevent damage of the target device due to mains supply fluctuations, communication errors or if the PC operating system fails. In the event of such errors the LabProg+ performs independently of the PC exactly specified special sequences so that the target device remains intact. LabProg+ performs **selftest** (diagnostic tests), including

verification of pin-driver voltage/level, for accurate timing of the signals applied to the target device and for reliable communication with the PC.

LabProg+ incorporates optimal PCB design criteria to minimize unwanted effects at the pins of the target socket (such as ground-bouncing and supply/programming voltage glitches). All the inputs of the LabProg+, including the socket, are **protected against ESD** and whilst inserted the target device is also protected against ESD damage.

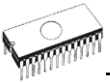
LabProg+ performs programming **verification** at the marginal level of supply voltage, which, obviously, improves programming yield and guarantees long data retention.

The LabProg+ is driven by an **easy-to-use**, **control program** with pull-down menus, hot keys and on-line help. You can select the target device by its class, by manufacturer or simply by typing a fragment of vendor name or part number. Besides the standard device-related commands (blank check, read, program, verify) have been implemented some test functions (illegal bit test, insertion test), and some special functions (production mode - start the action immediately after insertion of device into ZIF socket). The control program permits data manipulation within the buffer and between the buffer and associated files, and performs automatic file format detection and conversion.

The control program provides an **auto-increment function** that enables you to assign individual serial numbers to each programmed device, this function simply increments a serial number in the buffer each time a new device is inserted in the socket. Furthermore, the function enables the operator to read serial numbers and/or any programmed device identification signatures from a file.

Various **socket converters** are available for the LabProg+, DIL to PLCC, DIL to PSOP, DIL to SOIC, DIL to TQFP and DIL to TSOP. Software controlled pin-swapping capability together with universal converters are what give the LabProg+ the ability to program the target device in any package type. For example, just one DIL44 to PLCC44 converter will allow the LabProg+ to handle any PLCC44 target device in the current device list. Most devices with more than 48 pins require a simple adapter usually available from ELNEC or easily built by the user from information provided in the file "adapters.txt" included with the control program.

It is important to remember that in most cases new devices require **only a software upgrade** since the LabProg+ has 48 true pin drivers, which can perform as required under program

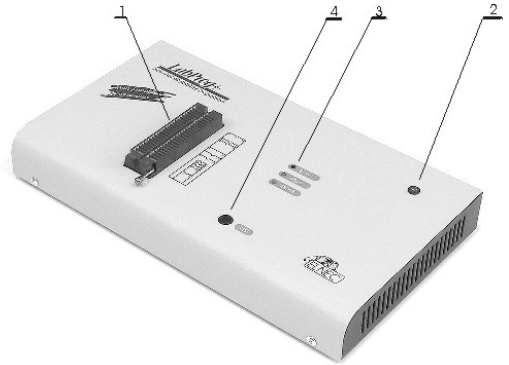


control. With our prompt service new devices can be added to the current list within hours!

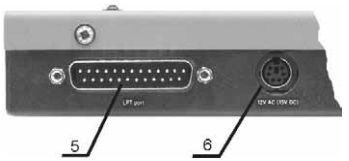
Advanced design, including protection circuits, original brand components and careful manufacturing allows us to provide a **three-year warranty** on parts and labor for the LabProg+ (limited 25,000-cycle warranty on ZIF socket).

LabProg+ elements

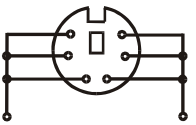
① 48 pin ZIF (**Z**ero **I**nsertion **F**orce) socket
② LED indicator power/sleep
③ LED indicators for work result
④ YES! Button



⑤ Connector for PC <-> LabProg+ communication cable
⑥ Power supply connector

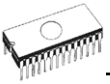


Power supply connector



AC	~	~	12VAC/1A
DC	+	-	15VDC/1A
DC	-	+	15VDC/1A

Note: *When the LabProg+ programmer isn't active, control program switch the programmer into "sleep mode". In this state it consumes less than 1W of power. Therefore the programmer has no On/Off switch. When the power LED indicator glows with a low intensity the LabProg+ is in sleep mode.*



Connecting LabProg+ to the PC

Switch off PC and programmer. Insert the communication cable included with your LabProg+ programmer package to a free printer port on your PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counter-connectors. This is very important. It may be uncomfortable to switch between printer cable and programmer cable, though it is not recommended to operate the LabProg+ programmer through a mechanical printer switch. Use of an electronic printer switch is impossible. But you can install a second multi-I/O in your computer, thus obtaining a supplementary printer port, says LPT2. So your printer may remain on LPT1 while the programmer on LPT2.

Switch on the PC.
Connect the mains connector of the power supply (or the wall-plug power supply itself) to a mains plug, then connect the mini-DIN connector to the programmer's connector labeled "12VAC". At this time all 'work result' LEDs (and 'POWER' LED) light up successive and then switch off. Once the POWER LED lights with low brightness then the LabProg+ programmer is ready to run.
Next run the control program for LabProg+.

Note: *When the PC is switch off and you turn on programmer, LED maybe not blinking, before programmer maybe permanent on reset.*

Caution! *If you don't want to switch off your PC when connecting the LabProg+, proceed as follows*:

- ***When connecting** the programmer to the PC: **FIRST** insert the **communications cable** and **THEN** the **power-supply connector.***
- ***When disconnecting** the programmer from the PC: **FIRST** disconnect the **power-supply connector** and **THEN** the **communication cable**.*

From LabProg+'s point of view the connecting and disconnecting sequence is irrelevant. Protection circuits on all programmer inputs keep it safe. **But think of your PC please**.

Note: *Programmer's protection electronics protect the target device and the programmer itself against either short or long-term power failures and, partly, also against a PC failure. However, it is not possible to grant the integrity of the target device due to incorrect, user-selected programming parameters. Target device may be not destroyed by forced*

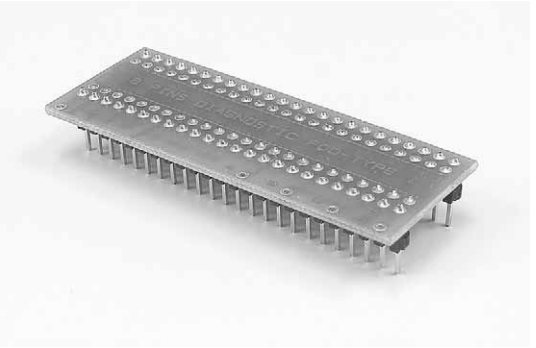
interruption of the control program (reset or switch-off PC), by removing the physical connection to the programmer, but the content of actually programmed cell may remains undefined. Don't unplug the target device from the ZIF socket during work with devices (LED BUSY shine).

Problems related to the LabProg+ ⇔ PC interconnection, and their removing

If you have any problems with LabProg+ ⇔ PC interconnection, see section ***Common notes*** please.

Selftest and Calibration

If you feel that your programmer does not react according to your expectation, please run the programmer selftest using Diagnostic POD, enclosed with the standard delivery package. For optimal results with programmer we recommend you undertake every 6 months, an extended test and to check the calibration. See instructions for selftest in the **Diagnostics** menu of PG4UW.



Technical specification

Socket, pin drivers and DACs

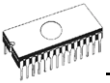
- 48-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 48-pin
- Three D/A converters for VCCP, VPP1, and VPP2, with controllable rise and fall time and current limitation
- TTL driver provides H, L, CLK, pull-up, pull-down, or tri-state on all 48 pins
- full support of Low Voltage circuits from 1.8 V up
- autocalibration

Device support

Programmer

- EPROM: NMOS/CMOS, 1702*, 2708*, 27xxx and 27Cxxx series, from 16Kbit to 32Mbit with 8/16 bit data width, full support for LV series
- EEPROM: NMOS/CMOS, 28xxx, 28Cxxx, 27EExxx series, with 8 / 16 bit data width
- Flash EPROM: 28Fxxx, 29Cxxx, 29Fxxx, 29BVxxx, 29LVxxx, 29Wxxx, 49Fxxx series, from 256Kbit to 32Mbit, with 8 / 16 bit data width, full support for LV series
- serial E(E)PROM: 17Cxxx, 18Vxxx, 24Cxxx, 24Fxxx, 25Cxxx, 45Dxxx, 59Cxxx, 85xxx, 93Cxxx, EPCxxx, NVM3060, MDAxxx series, full support for LV series
- PROM: Harris, NS, Philips/Signetics, Tesla, TI
- NV RAM: Dallas DSxxx, SGS/Inmos MKxxx, SIMTEK STKxxx, XICOR 2xxx, ZMD U63x, Benchmarq BQxxx, Catalyst (CSI) CAT22Cxx, Greenwitch Instr. GRxxx, Ramtron FMxxxx, Seiko Instruments S-22xxx, SGS-Thomson (STM) Mxxxxx series
- PLD: SPLD, CPLD: Altera, Atmel, AMD - Vantis, Lattice, NS, Philips, TI, Xilinx
- microcontrollers 48 series: 87x41, 87x42, 87x48, 87x49, 87x50 series
- microcontrollers 51 series: 87xx, 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx, Philips 87C748..752, 87LPCxxx series
- microcontrollers Microchip PIC: PIC12Cxxx, PIC12Fxxx, PIC16C5x, PIC16Cxxx, PIC16Fxxx, PIC17Cxxx, PIC18Cxxx series
- microcontrollers Atmel AVR: AT90Sxxxx, ATtiny, ATmega series
- microcontrollers NEC (uPD78Pxxx series), SGS-Thomson (ST6 series), Zilog (Z8 series), Motorola (68HC05, 68HC11), NS (COP8), Holtek, Samsung, Scenix, Fujitsu, Hitachi, TI…

Note: *Devices marked * are obsolete, programming with additional module*



I.C. Tester

- TTL type: 54,74 S/LS/ALS/H/HC/HCT series
- CMOS type: 4000, 4500 series
- static RAM: 6116 .. 624000
- user definable test pattern generation

Package support

- package support includes DIP, PLCC, SDIP, SOIC, PSOP, TSOP, TQFP and other (ask for package converters)
- support all devices in DIP with default ZIF-48 socket
- support devices in non-DIP packages up to 48 pin with universal adapter (optional accessory, to be ordered separately)
- compatible with third-party adapters for non-DIP support

Programming speed

Note. *These times strongly depend on PC speed, LPT port type and operating system free resources. Therefore are given values of two different PC configurations for comparison.*

Device	Operation	Time A	Time B
27C010	programming and verify	39 sec	32 sec
AT29C040A	programming and verify	110 sec	90 sec
AM29F040	programming and verify	165 sec	139 sec
PIC16C67	programming and verify	46 sec	41 sec

Time A conditions: *Pentium MMX, 250 MHz, ECP/EPP, WIN98.*
Time B conditions: *Pentium III, 500 MHz, ECP/EPP on PCI bus, WIN98.*

Device operations

- **standard:**
  - automatic ID-based selection of EPROM/Flash EPROM
  - blank check
  - read
  - program
  - verify
  - erase
  - configuration and security bit program
  - illegal bit test
  - checksum
- **security**
  - insertion test
  - contact check
  - ID byte check

- **special**
  - production mode (automatic start immediately after device insertion)
  - automatic device serial number incrementation
  - statistics
  - count-down

Buffer operations

- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Supported file formats

- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-space-HEX
- POF (Altera), JEDEC (ver. 3.0.A), for example from ABEL, CUPL, PALASM, TANGO PLD, OrCAD PLD, PLD Designer ISDATA etc.

PC system requirements

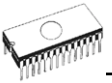
See section ***Introduction/ PC requirements***

General

- operating voltage 12..15V AC/max.1A
- power consumption - max. 12W in active mode, approx. 1W sleep mode
- dimensions 275x157x47 mm
- weight (without external adapter) 1.5 kg
- operating temperature 5°..40°C
- humidity 20%.80%, non condensing

Package included

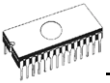
- LabProg+ programmer
- connection cable
- diagnostic POD for selftest
- anti-dust cover to ZIF socket
- power supply adapter
- user manual
- software
- "DEVICE PROBLEM REPORT" form
- registration card
- transport case



Additional services

- Keep Current
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.

SmartProg



Introduction

SmartProg is next member of new generation of Windows 95/98/Me/NT/2000/XP based ELNEC universal programmers. Programmer is built to meet the demands of the development labs and field engineers to universal, but portable programmer.

SmartProg is a small, fast and powerful programmer of all kinds of programmable devices. Using build-in in-circuit serial programming (ISP) connector the programmer is able to program ISP capable chips in-circuit.

Provides very competitive price but excellent hardware design for reliable programming. Nice "value for money" in this class.

Very fast programming due to high-speed FPGA driven hardware and support of IEEE1284 (ECP/EPP) high-speed parallel port. Surely faster than competitors in this category.

SmartProg interfaces with the IBM PC, AT or above, portable or desktop personal computers through any standard parallel (printer) port (no special interface card needed). Therefore you can take programmer and move it to another PC without assembly/disassembly of PC.

SmartProg has 40 powerful TTL pindrivers provide H/L/pull_up/pull_down and read capability for each pin of socket. Advanced pindrivers incorporate high-quality high-speed circuitry to deliver signals without overshoot or ground bounce for all supported devices. Pin drivers operate down to 1.8V so you'll be ready to program the full range of today's advanced low-voltage devices.

The programmer performs **device insertion test** (wrong device position in socket) and contact check (poor contact pin-to-socket) before it programs each device. These capabilities, supported by signature-byte check help prevent chip damage due to operator error.

Programmer's hardware offers enough resources for **selftest**, that control program is any time be able to check pindrivers, present and correct level of all voltages, check the timing and communication between programmer and PC.

SmartProg programmer performs programming **verification** at the marginal level of supply voltage, which, obviously, improves programming yield, and guarantees long data retention.

SmartProg is driven by an easy-to-use control program with pull-down menus, hot keys and on-line help. Selecting of device is performed by its class, by manufacturer or simply by typing a fragment of vendor name and/or part number. Standard device-related commands (read, blank check, program, verify, erase) are enhanced by some test functions (insertion test, signature-byte check), and some special functions (autoincrement). All known data formats are supported. Automatic file format detection and conversion during load of file.

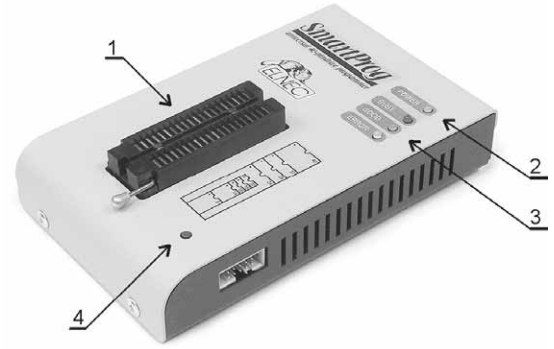
The rich-featured **autoincrement function** enables to assign individual serial numbers to each programmed device - or simply increments a serial number, or the function enables to read serial numbers or any programmed device identification signatures from a file.

Various socket converters are available to handle device in PLCC, SOIC and other packages.

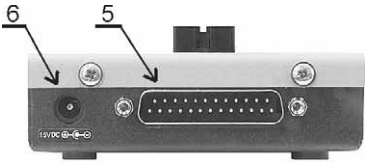
Advanced design, including protection circuits, original brand components and careful manufacturing allows us to provide a **three-year warranty** on parts and labor for the SmartProg (limited 25,000-cycle warranty on ZIF socket).

SmartProg elements

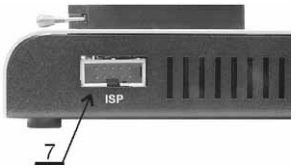
① 40 pin ZIF socket
② LED power/sleep
③ LED, which indicate work result
④ YES! button



⑤ Connector for PC ⇔ SmartProg communication cable
⑥ Power supply connector



⑦ Connector for ISP



Note: *Due to low power consumption of SmartProg in inactive state, it doesn't require power switch*.

Connecting SmartProg to PC

Switch off the PC and programmer. Insert the connection cable, included in the SmartProg programmer delivery, to the free printer port of PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counterconnectors. This is very important mainly for the connector to programmer. Though replacing the printer cable by the programmer cable is uncomfortable, it is not recommended to operate the SmartProg programmer through a mechanical printer switch. Use of an electronic printer switch isn't possible.

Connect the mains connector of the power supply (or wall-plug power supply self) to a mains plug, connect the connector to the programmer's connector labeled 15VDC. At this time all 'work result' LEDs (and 'POWER' LED) light up successive and then switch off. Once the POWER LED lights with low brightness then the SmartProg is ready to run.

Next switch on the PC and run the control program.

Caution! *If you don't want to switch off your PC when connecting the SmartProg, proceed as follows*:

- ***When connecting** the programmer to the PC: **FIRST** insert the **communications cable** and **THEN** the **power-supply connector.***
- ***When disconnecting** the programmer from the PC: **FIRST** disconnect the **power-supply connector** and **THEN** the **communication cable.***

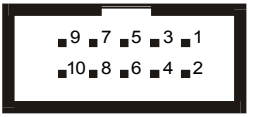
Problems related to the SmartProg ⇔ PC interconnection, and their removing

If you have any problems with SmartProg ⇔ PC interconnection, see section ***Common notes*** please.

In-system serial programming by SmartProg

For general definition, recommendation and direction about ISP see section ***Common notes / ISP*** please.

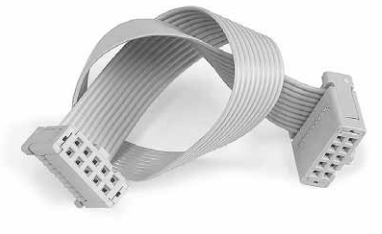
Description of SmartProg ISP connector



Pin	Description
1	VCCP for target device, with sense
2, 10	H/L/read, GND, VCCP, VPP
3	H/L/read, GND, VPP
4	H/L/, read, VPP
5	NC
6	H/L/read, GND, VCCP
7,9	GND
8	H/L/read, GND

Specification of ISP connector pins depends on programming device and is displayed at **Additional info window**. These specifications correspond with application notes published of device manufacturers. Used application notes you may find on **www.elnec.com,** section application notes.

Note: *Pin no. 1 is signed by triangle scratch on ISP cable connectors*.



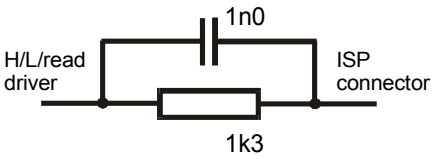
SmartProg ISP cable

Warnings:

- ***When** you use SmartProg as **ISP programmer**, **don't insert** device to **ZIF socket**.*
- ***When** you programm devices in **ZIF socket**, **don't insert** ISP cable to **ISP connector**.*
- *Use only **attached ISP cable**. When you use other ISP cable (other material, length…), programming may occur unreliable.*

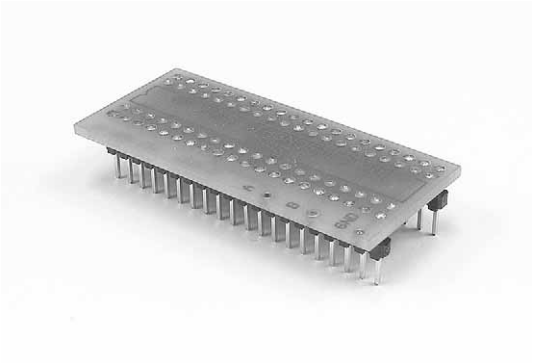
- ***SmartProg can supply** programmed device only, but target system **cannot supply SmartProg**.*
- *SmartProg apply programming voltage to target device and checks his value (target system can modify programming voltage). If the programming voltage is different as expected, no action with target device will be executed.*

Note: *H/L/read SmartProg driver*



Selftest and calibration

If you have feeling, that your programmer doesn't behave according to yours expectations (but at least every 3 month), please run selftest using Diagnostic POD, that contains standard programmer delivery. For optimal fruit of work with SmartProg programmer is recommended every 6 months (no more) to make extended test and check the calibration. All this activities you can find in **Diagnostics** menu of PG4UW.



Manipulation with the programmed device

After selection of desired device for your work, you can insert into the open ZIF socket (the lever is up) and close socket (the lever is down). The correct orientation of the programmed device in ZIF socket is shown on the picture near ZIF socket on the programmer's cover. The programmed device is necessary to insert into the socket also to remove from the socket when LED BUSY light off.

Technical specification

HARDWARE

Programmer

- two D/A converters for VCCP and VPP, controllable rise and fall time
- VCCP range 0..7V/350mA
- VPP range 0..25V/200mA
- FPGA based IEEE 1284 slave printer port, up to 1MB/s transfer rate
- autocalibration
- selftest capability

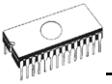
ZIF socket, pindriver

- 40-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 40-pins
- pindriver: 40 TTL pindrivers, universal GND/VCC/VPP pindriver
- FPGA based TTL driver provides H, L, CLK, pull-up, pull-down on all pindriver pins, level H selectable from 1.8 V up to 5V
- in-circuit serial programming (ISP) capability included
- continuity test: each pin is tested before every programming operation

DEVICE SUPPORT

Programmer

- EPROM: NMOS/CMOS, 2708 (*2), 27xxx and 27Cxxx series, with 8/16 bit data width, full support of LV series (*1)
- EEPROM: NMOS/CMOS, 28xxx, 28Cxxx, 27EExxx series, with 8/16 bit data width, full support of LV series (*1)
- Flash EPROM: 28Fxxx, 29Cxxx, 29Fxxx, 29BVxxx, 29LVxxx, 29Wxxx, 49Fxxx series, with 8/16 bit data width, full support of LV series (*1)
- Serial E(E)PROM: 17Cxxx, 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx, series, full support of LV series
- Configuration PROM: 17xxx, LV series including
- NV RAM: Dallas DSxxx, SGS/Inmos MKxxx, SIMTEK STKxxx, XICOR 2xxx, ZMD U63x series
- PLD: series: Atmel, AMD-Vantis, Cypress, ICT, Lattice, NS (*1)
- microcontrollers MCS51 series: 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx series from Atmel, Atmel W&M, Intel, Philips, SST, Winbond, (*1*2), parallel and serial (ISP) mode



- microcontrollers Atmel AVR: ATtiny, AT90Sxxx, ATmega series, (*1*2), parallel and serial (ISP) mode
- Microcontrollers Microchip PICmicro: PIC12Cxxx, PIC16C5x, PIC16Cxxx, PIC17Cxxx, PIC18Cxxx series, 8-40 pins (*1*2), parallel and serial (ISP) mode
- Microcontrollers Scenix (Ubicom): SX18xxx, SX20xxx, SX28xxx series

Notes:

*(*1) - suitable adapters are available for non-DIL packages*

*(*2) - there exist only few adapters for devices with more than 40 pins. Therefore think please about more powerful programmer (LabProg+, JetProg), if you need to program devices with more than 40 pins*

Look please at list of all supported devices.

I.C. Tester

- Static RAM: 6116 .. 624000

Programming speed

Note. *These times strongly depend on PC speed, LPT port type and operating system free resources. Therefore values of two different PC configurations are given for comparison.*

Device	Operation	Time A	Time B
27C010	programming and verify	28 sec	24 sec
AT29C040A	programming and verify	38 sec	28 sec
AM29F040	programming and verify	102 sec	87 sec
PIC16C67	programming and verify	13 sec	11 sec
PIC18F452	programming and verify	11 sec	9 sec
AT89C52	programming and verify	18.5 sec	16.5 sec

Time A conditions: *Pentium MMX, 250 MHz, ECP/EPP, WIN98.*

Time B conditions: *Athlon, 750 MHz, ECP/EPP on PCI bus, WIN98.*

SOFTWARE

- **Algorithms:** only manufacturer approved or certified algorithms are used. Custom algorithms are available at additional cost.
- **Algorithm updates:** software updates are available approx. every 2 weeks, free of charge.
- **Main features:** revision history, session logging, on-line help, device and algorithm information

Device operations

- **standard:**
  - intelligent device selection by device type, manufacturer or typed fragment of part name
  - blank check, read, verify
  - program
  - erase
  - configuration and security bit program
  - illegal bit test
  - checksum
- **security**
  - insertion test
  - contact check
  - ID byte check
- **special**
  - auto device serial number increment
  - statistic
  - count-down mode

Buffer operations

- view/edit, find/replace
- fill, copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

File load/save

- no download time because programmer is PC controlled
- automatic file type identification

Supported file formats

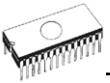
- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S-record, MOS, Exormax, Tektronix, ASCII-SPACE-HEX

PC system requirements

See section ***Introduction/ PC requirements***

GENERAL

- operating voltage 15V DC, max. 0.5A (adapter included in package)
- power consumption max. 6W active, about 0.5W sleep
- dimensions 160x95x35 [mm] (6.3x3.8x1.4 [inch])
- weight (without external power adapter) ca. 500g
- temperature 5°..40°C
- humidity 20%..80%, non condensing



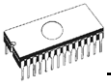
Base configuration package includes

- SmartProg programmer
- connection cable PC-programmer
- ISP cable
- diagnostic POD for selftest
- anti-dust cover for ZIF socket
- wall plug adapter 220V to 15V DC/500mA, unstabilized
- user manual
- software
- "DEVICE PROBLEM REPORT" form
- registration card
- transport case

Additional services

- Keep Current
- AlgOR
- free technical support (hot line)
- free life-time software update via Internet

PREPROM-02aLV



Introduction

PREPROM-02aLV is a small and powerful EPROM, EEPROM, Flash EPROM and serial EEPROM **programmer** and static RAM **tester**, designed for professional mobile applications. In addition, PREPROM-02aLV programmer with auxiliary modules support also microprocessors (MCS48, MCS51, PICmicro, AVR), GALs, etc. Generators for supply voltage and programming voltage are digitally controlled and level of H can be limited, therefore programmer can work with 'true LV' device too - from 2V. Performance, dimensions and speed of PREPROM-02aLV can be used both in maintenance and in production.

PREPROM-02aLV interfaces with the IBM PC, AT or higher, portable or desktop personal computers. Programmers allow you to directly connect to your PC through any standard parallel (printer) port - no special interface card is needed.

Built-in protection circuits eliminate damage of programmed device due to mains supply error, communication error or if PC is frozen. Programmer's hardware afford enough resources for selftest, that control program is any time be able to check pindrivers, present of all voltages, check the timing and communication between programmer and PC.

Programming verification is performed by the programmer at the marginal level of VCCP, which, obviously, improves programming yield, and guarantees long data retention.

The advantage of PREPROM-02aLV is its easy-to-use comfortable software with pull-down menu, hot keys and on-line help. You can select device by classes, by manufacturers or simply type device vendor's name and part number. The standard device operation functions (read, blank check, program, verify) have been completed with some test functions (ID check). The program facilitates the use of the buffer and files, including automatic file format detection and conversion.

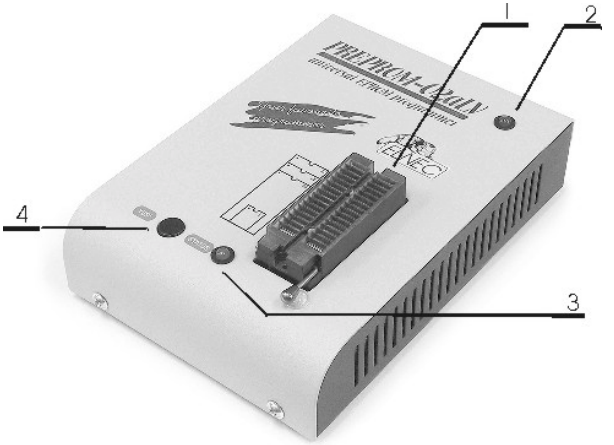
The control program provides an **auto-increment function** that enables you to assign individual serial numbers to each programmed device, this function simply increments a serial number in the buffer each time a new device is inserted in the socket. Furthermore, the function enables the operator to read serial numbers and/or any programmed device identification signatures from a file.

The PREPROM-02aLV can make for use of DIL to PLCC, SOIC, PSOP socket converters.

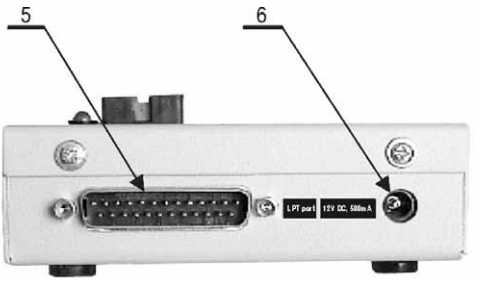
Taking into consideration the PREPROM-02aLV programmer design, protective circuits, and the components used, the manufacturer is able to provide a three-year warranty on parts and labor for the programmer (limited 25,000-cycle warranty on the ZIF socket).

PREPROM-02aLV elements

① 32 pin ZIF socket
② LED power/sleep
③ LED, which indicate work result
④ YES! button



⑤ communication connector, for PC ⇔ PREPROM-02aLV cable connection
⑥ connector for power supply connection



Note: *Due to low power consumption of PREPROM-02aLV in inactive state, it doesn't require power switch.*

Connecting PREPROM-02aLV programmer to PC

Switch off the PC and programmer. Insert the connection cable, included in the PREPROM-02aLV programmer delivery, to the free printer port of PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counterconnectors. This is very important mainly for the connector to programmer. Though replacing the printer cable by the programmer cable is uncomfortable, it is not recommended to operate the PREPROM-02aLV programmer through a mechanical printer switch. Use of an electronic printer switch isn't possible.

Connect the mains connector of the power supply (or wall-plug power supply self) to a mains plug, connect the connector to the programmer's connector labeled 12VDC. Then, on the programmer lights up LED POWER and the programmer PREPROM-02aLV is ready to run. Next switch on the PC and run the control program.

Caution! *If you don't want to switch off your PC when connecting the PREPROM-02aLV, proceed as follows:*

- ***When connecting** the programmer to the PC: **FIRST** insert the **communications cable** and **THEN** the **power-supply connector.***
- ***When disconnecting** the programmer from the PC: **FIRST** disconnect the **power-supply connector** and **THEN** the **communication cable**.*

Problems related to the PREPROM-02aLV ⇔ PC interconnection, and their removing

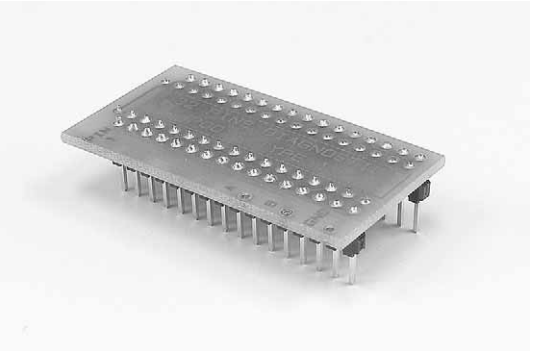
If you have any problems with PREPROM-02aLV ⇔ PC interconnection, see section ***Common notes*** please.

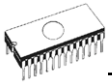
Note for laptop and PREPROM-02aLV users

Since the PREPROM-02aLV programmer is supplied with 12V, notebook and/or laptop users may operate their PREPROMs-02aLV everywhere this voltage is available, e.g. in a car. Ten series-connected NiCd batteries can supply the programmer for more than one hour at full performance. Unwanted swapping has no effects - the programmer is protected against the reversal of the power supply poles. The supply cable you can make yourself, polarity is:

Selftest and calibration

If you have feeling, that your programmer doesn't behave according to yours expectations (but at least every 3 month), please run PREPROM-02aLV selftest using Diagnostic POD, that contains standard programmer delivery. For optimal fruit of work with PREPROM-02aLV programmer is recommended every 6 months (no more) to make extended test and check the calibration. All this activities you can find in **Diagnostics** menu of PG4UW.





PREPROM-02aLV specification

Socket, pin drivers and DACs

- 32-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 32-pin
- two D/A converters for VCCP and VPP, with controllable rise/fall time and current limitation
- TTL driver provides H, L and read all 32 pins
- full support of Low Voltage circuits from 2.0 V

Device support

Programmer

- EPROM: NMOS/CMOS, 27xxx and 27Cxxx series, with 8/16* bit data width
- EEPROM: NMOS/CMOS, 28xxx, 28Cxxx, 27EExxx series, with 8/16* bit data width
- Flash EPROM: 28Fxxx, 29Cxxx, 29Fxxx, 29BVxxx, 29LVxxx, 29Wxxx, 49Fxxx series, from 256Kbit to 32Mbit, with 8/16* bit data width
- serial E(E)PROM: 17Cxxx*, 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx, NVM3060, MDAxxx* series
- NV RAM: Dallas DSxxx, SGS/Inmos MKxxx series
- PLD*: AMD PALCE, GALs, PEELs series
- microcontrollers 48 series*: 87x41, 87x42, 87x48, 87x49, 87x50 series
- microcontrollers 51 series*: 87xx, 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx, Philips 87C748..752 series
- microcontrollers Microchip PIC*: PIC12Cxxx, PIC16C5x, PIC16Cxxx, PIC18Cxxx series
- microcontrollers Atmel AVR*: AT90Sxxxx series
- microcontrollers NEC*: uPD78Pxxx series
- note: * - programming with additional module

I.C. Tester

- static RAM: 6116 .. 624000

Programming speed

- M27C256, programming + verifying, PC486/DX50 : 16 sec
- M27C010, programming + verifying, PC586/100 : 40 sec

Device operations

- **standard:**
  - automatic ID-based selection of EPROM/Flash EPROM
  - blank check
  - read
  - program
  - verify
  - erase
  - configuration and security bit program
  - illegal bit test
- **security:**
  - contact check
  - ID byte check
- **special**
  - auto device serial number increment

Buffer operations

- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Supported file formats

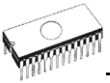
- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-space-HEX
- JEDEC

PC system requirements

See section ***Introduction/ PC requirements***

General

- operating voltage 12VDC/max. 500mA (adapter 220VAC/12VDC included in package)
- power consumption 6W max.
- dimensions 160x110x50 [mm] (6.3x4.3x2.0 [inch])
- weight (without external adapter) ca. 650g
- temperature 5° ÷ 40°C
- humidity 20%..80%, non condensing



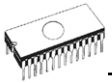
Package included

- PREPROM-02aLV programmer
- 1.8m (6 ft.) connecting cable PREPROM-02aLV to (IBM) PC
- wall plug adapter 220V/50Hz to 12V DC
- diagnostic POD, that enable programmer's selftest
- anti-dust cover to ZIF socket
- software
- user manual
- "DEVICE PROBLEM REPORT" form
- registration card
- transport packing

Additional services

- Keep Current
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.

T51prog



Introduction

T51prog is a new generation of Windows 95/98/ME/NT/2000 based ELNEC specialized programmers. Programmer is capable to support all today available microcontrollers of MCS51 series (up to 40 pins) and AVR microcontrollers (8-40 pins) by parallel and serial way. T51prog has been developed in close cooperation with Atmel W&M., therefore programmer's hardware is focused to support all current and future microcontrollers of Atmel W&M MCS51 family.

T51prog is little, very fast and powerful portable programmer for MCS51 series and Atmel AVR microcontrollers. T51prog enables also programming of serial EEPROM with IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx) interface types. Using build-in in-circuit serial programming (ISP) connector programmer is capable to program MCS51 family microcontrollers and Atmel AVR microcontrollers in serial way.

Provides very competitive price but excellent hardware design for reliable programming. Nice "value for money" in this class.

Very fast programming due to high-speed FPGA driven hardware and support of IEEE1284 (ECP/EPP) high-speed parallel port. Surely faster than competitors in this category.

T51prog interfaces with the IBM PC, AT or higher, portable or desktop personal computers. A programmer allows you to directly connect to your PC through any standard parallel (printer) port - no special interface card is needed.

T51prog has 40 powerful TTL pindrivers provide H/L/pull_up/pull_down and read capability for each pin of socket. Advanced pindrivers incorporate high-quality high-speed circuitry to deliver programming without overshoot or ground bounce for all supported devices. Pin drivers operate down to 1.8V so you'll be ready to program the full range of today's advanced low-voltage devices.

The programmer performs **device insertion test** (wrong or backward position) and contact check (poor contact pin-to-socket) before it programs each device. These capabilities, supported by signature-byte check help prevent chip damage due to operator error.

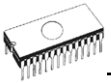
Programmer's hardware offers enough resources for **selftest**, that control program is any time be able to check pindrivers, present and correct level of all voltages, check the timing and communication between programmer and PC.

T51prog performs programming **verification** at the marginal level of supply voltage, which, obviously, improves programming yield, and guarantees long data retention

T51prog is driven by an easy-to-use comfortable software with pull-down menu, hot keys and on-line help. You can select device by classes, by manufacturers or simply type device vendor's name and part number. The standard device operation functions (read, blank check, program, verify) have been completed with some test functions. The program facilitates the use of the buffer and files, including automatic file format detection and conversion.

The control program provides an **auto-increment function** that enables you to assign individual serial numbers to each programmed device, this function simply increments a serial number in the buffer each time a new device is inserted in the socket. Furthermore, the function enables the operator to read serial numbers and/or any programmed device identification signatures from a file.

For T51prog are available DIL to PLCC and SOIC socket converters.



Connecting T51prog programmer to PC

Switch off the PC and programmer. Insert the connection cable, included in the T51prog programmer delivery, to the free printer port of PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counterconnectors. This is very important mainly for the connector to programmer. Though replacing the printer cable by the programmer cable is uncomfortable, it is not recommended to operate the T51prog programmer through a mechanical printer switch. Use of an electronic printer switch isn't possible.

Connect the mains connector of the power supply (or wall-plug power supply self) to a mains plug, connect the connector to the appropriate programmer's connector. Then, on the programmer lights up LED POWER and the programmer T51prog is ready to run. Next switch on the PC and run the control program.

Caution! *If you don't want to switch off your PC when connecting the T51prog, proceed as follows*:

- ***When connecting*** *the programmer to the PC:* ***FIRST*** *insert the* ***communications cable*** *and* ***THEN*** *the* ***power-supply connector.***
- ***When disconnecting*** *the programmer from the PC:* ***FIRST*** *disconnect the* ***power-supply connector*** *and* ***THEN*** *the* ***communication cable.***

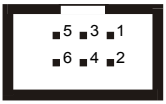
Problems related to the T51prog ⇔ PC interconnection, and their removing

If you have any problems with T51prog ⇔ PC interconnection, see section ***Common notes*** please.

In-System serial programming by T51prog

For general definition, recommendation and direction about ISP see section ***Common notes / ISP*** please.

Description of T51prog ISP connector



Pin	Description
1,3,4,5	H/L/read
2	target VCC sense
6	GND

Specification of ISP connector pins depends on programming device and is displayed at **Additional info window**. These specifications correspond with Atmel application note AVR910: In-System Programming. Used application note you may find on **www.elnec.com**, section application notes.

Note: *Pin no. 1 is signed by triangle scratch on ISP cable connectors*.

ISP cable of T51prog

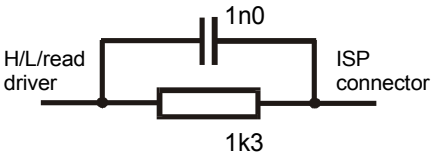
Warnings:

- ***When** you use T51prog as **ISP programmer**, **don't insert** device to **ZIF socket**.*
- ***When** you programm devices in **ZIF socket**, **don't insert** ISP cable to **ISP connector**.*

- *Use only **attached ISP cable**. When you use other ISP cable (other material, length…), programming may occur unreliable.*

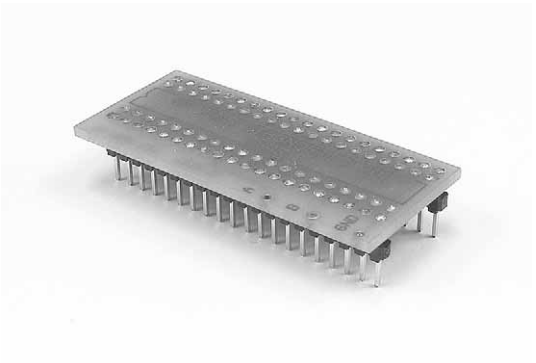
- ***T51prog cannot** supply target system and **target system cannot supply T51prog**. Before action with target device T51prog check power supply of target system. If this power supply is different as expected, no action with device will be executed.*

Note: *H/L/read T51prog driver.*



Selftest and calibration

If you have feeling, that your programmer doesn't behave according to yours expectations (but at least every 3 month), please run selftest using Diagnostic POD, that contains standard programmer delivery. For optimal fruit of work with T51prog programmer is recommended every 6 months (no more) to make extended test and check the calibration. All this activities you can find in **Diagnostics** menu of PG4UW.



Manipulation with the programmed device

After selection of desired device for your work, you can insert into the open ZIF socket (the lever is up) and close socket (the lever is down). The correct orientation of the programmed device in ZIF socket is shown on the picture near ZIF socket on the programmer's cover. The programmed device is necessary to insert into the socket also to remove from the socket when LED BUSY light off.

Warning! *T51prog programmer hasn't protection devices, which protect the content of programmed device against critical situations, for example power failures and PC failure (interrupted cable...). Moreover, an device is usually destroyed in the programming mode due to forced interruption of the control program run (Reset or switching the computer off) due to removing the connecting cable, or unplugging the programmed device from the ZIF socket. Incorrectly placed device in the ZIF socket can cause its damage or destruction.*

T51prog specification

Socket, pin drivers and DACs

- FPGA based IEEE 1284 slave printer port, up to 1MB/s transfer rate
- 40-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 40-pin
- two D/A converters for VCCP and VPP, controllable rise and fall time
- VCCP range 0 – 6.5V / 150mA
- VPP range 0 – 15V / 100mA
- special GND/VCCP/VPP pindriver for MCS51 and AVR devices
- spare GND, VCCP and VPP driver, which add additional made-by-wire GND/VCCP/VPP pin capability for future devices
- FPGA based TTL driver provides H, L, CLK, pull-up, pull-down on and read for all pindriver pins,
- level H selectable from 1.8 V up to 5V
- in-circuit serial programming (ISP) capability included

Device support

- microcontrollers 51 series: 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx series from Atmel, Atmel W&M, Intel, Philips, ... 8/40 pins
- microcontrollers Atmel AVR: ATtiny, AT90Sxxx, ATmega series (parallel and serial mode), 8-40 pins
- serial E(E)PROM: 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx series

Programming speed

Note. *These times strongly depend on PC speed, LPT port type and operating system free resources. Therefore are given values of two different PC configurations for comparison.*

Device	Operation	Time A	Time B
AT89C52	programming and verify	17,5 sec	15,5 sec
T87C5111	programming and verify	45 sec	25 sec

Time A conditions: *Pentium MMX, 250 MHz, ECP/EPP, WIN98.*

Time B conditions: *Athlon, 750 MHz, ECP/EPP on PCI bus, WIN98.*

Device operations

- **standard:**
  - intelligent device selection by device type, manufacturer or typed fragment of part name
  - blank check
  - read
  - program
  - verify
  - erase
  - configuration and security bit program
- **security:**
  - insertion test, reverse insertion check
  - contact check
  - ID byte check
- **special:**
  - statistic
  - count-down mode
  - auto device serial number increment

Buffer operations

- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Supported file formats

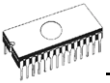
- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-space-HEX, Tektronix

PC system requirements

See section ***Introduction/ PC requirements***

General

- operating voltage 12..15V DC/500mA
- power consumption 5W max.
- dimensions 132x66x30 [mm] (5.2x2.6x1.2 [inch])
- weight (without external adapter) ca. 200g
- temperature 5° ÷ 40°C
- humidity 20%..80%, non condensing



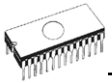
Package included

- T51prog programmer
- connection cable PC-programmer
- ISP cable
- diagnostic POD for selftest
- anti-dust cover for ZIF socket
- wall plug adapter 220V to 12V DC/500mA, unstabilized
- user manual
- software
- “DEVICE PROBLEM REPORT" form
- registration card
- transport case

Additional services

- Keep Current
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.

51&AVRprog



Introduction

51&AVRprog is little and powerful portable programmer for MCS51 series and Atmel AVR microcontrollers. 51&AVRprog enables also programming serial EEPROM with interface types IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx). The programmer is equipped by DIL ZIF 40pin socket. The quality of programmer is completed by comfortable control program.

51&AVRprog interfaces with the IBM PC, AT or higher, portable or desktop personal computers. A programmer allows you to directly connect to your PC through any standard parallel (printer) port - no special interface card is needed.

The advantage of 51&AVRprog is its easy-to-use comfortable software with pull-down menu, hot keys and on-line help. You can select device by classes, by manufacturers or simply type device vendor's name and part number. The standard device operation functions (read, blank check, program, verify) have been completed with some test functions. The program facilitates the use of the buffer and files, including automatic file format detection and conversion.

The 51&AVRprog can make for use of DIL to PLCC and SOIC socket converters.

Warning: *Jumper must be in position JP2-JP3 for MCS51 devices and JP3-JP4 for AVR family device. Jumper is accessible after removing bottom part of programmer case. Default jumper position is JP2-JP3 (for MCS51 family).*

Connecting 51&AVRprog programmer to PC

Switch off the PC and programmer. Insert the connection cable, included in the 51&AVRprog programmer delivery, to the free printer port of PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counterconnectors. This is very important mainly for the connector to programmer. Though replacing the printer cable by the programmer cable is uncomfortable, it is not recommended to operate the 51&AVRprog programmer through a mechanical printer switch. Use of an electronic printer switch isn't possible.

Connect the mains connector of the power supply (or wall-plug power supply self) to a mains plug, connect the connector to the appropriate programmer's connector. Then, on the programmer lights up LED POWER and the programmer 51&AVRprog is ready to run. Next switch on the PC and run the control program.

Caution! *If you don't want to switch off your PC when connecting the 51&AVRprog , proceed as follows*:

- ***When connecting** the programmer to the PC: **FIRST** insert the **communications cable** and **THEN** the **power-supply connector.***
- ***When disconnecting** the programmer from the PC: **FIRST** disconnect the **power-supply connector** and **THEN** the **communication cable.***

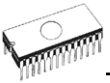
Problems related to the 51&AVRprog ⇔ PC interconnection, and their removing

If you have any problems with 51&AVRprog ⇔ PC interconnection, see section ***Common notes*** please.

Manipulation with the programmed device

After selection of desired device for your work, you can insert into the open ZIF socket (the lever is up) and close socket (the lever is down). The correct orientation of the programmed device in ZIF socket is shown on the picture near ZIF socket on the programmer's cover. The programmed device is necessary to insert into the socket also to remove from the socket when LED BUSY light off.

Warning! *51&AVRprog programmer hasn't protection devices, which protect the content of programmed device against critical situations, for example power failures and PC failure*



(interrupted cable...). Moreover, an device is usually destroyed in the programming mode due to forced interruption of the control program run (Reset or switching the computer off) due to removing the connecting cable, or unplugging the programmed device from the ZIF socket. Incorrectly placed device in the ZIF socket can cause its damage or destruction.

51&AVRprog specification

Socket, pin drivers and DACs

- 40-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 40-pin
- specialized GND/VCC/VPP pindriver for MCS51 and AVR devices
- TTL driver provides H, L and read for all needed pins

Device support

- microcontrollers 51 series: 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx series from Atmel, Intel, Philips, ... 20/40 pins
- microcontrollers Atmel AVR: AT90Sxxxx series (parallel mode), 8/20/40 pins
- serial E(E)PROM: 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx series

Note:
Serial EEPROMs programming is performed little bit slowly. Hardware of 51&AVRprog programmer isn't capable to support some latest MCS51/AVR chips.

Device operations

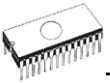
- **standard:**
  - blank check
  - read
  - program
  - verify
  - erase
  - configuration and security bit program
- **security:**
  - ID byte check
- **special:**
  - auto device serial number increment

Buffer operations

- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Supported file formats

- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-space-HEX, Tektronix



PC system requirements

See section ***Introduction/ PC requirements***

General

- operating voltage 15÷20V DC/200mA
- power consumption 3W max.
- dimensions 132x66x30 [mm] (5.2x2.6x1.2 [inch])
- weight (without external adapter) ca. 200g
- temperature 5° ÷ 40°C
- humidity 20%..80%, non condensing

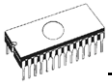
Package included

- 51&AVRprog programmer
- 1.3m (4.3 ft.) connecting cable 51&AVRprog to (IBM) PC
- wall plug adapter 15V DC/ 500mA, unstabilized
- software
- user manual
- "DEVICE PROBLEM REPORT" form
- registration card
- transport packing

Additional services

- Keep Current
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.

PIKprog+



Introduction

PIKprog+ is a member of the new generation of Windows 95/98/ME/NT/2000/XP based ELNEC specialized programmers. Programmer is capable to support all currently available Microchip™ PICmicro® series microcontrollers (8-40 pins) using parallel and serial algorithms. The PIKprog+ programmer isn't development grade programmer, but meets all Microchip's requirements to manufacturing grade of programmers. The PIKprog+ has been developed in close cooperation with Microchip™ company, therefore programmer's hardware is focused to support all current and future PICmicro® family microcontrollers.

PIKprog+ is a small, very fast and powerful portable programmer for PICmicro® family microcontrollers and serial EEPROM with IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx) interface types. Using build-in in-circuit serial programming (ISP) connector programmer is able to program PICmicro® family microcontrollers using serial algorithms.

Provides very competitive price but excellent hardware design for reliable programming. Nice "value for money" in this class.

Very fast programming due to high-speed FPGA driven hardware and support of IEEE1284 (ECP/EPP) high-speed parallel port. Surely faster than competitors in this category.

PIKprog+ interfaces with the IBM PC, AT or above, portable or desktop personal computers through any standard parallel (printer) port (no special interface card needed). Therefore you can take programmer and move it to another PC without assembly/disassembly of PC.

PIKprog+ has 40 powerful TTL pindrivers provide H/L/pull_up/pull_down and read capability for each pin of socket. Advanced pindrivers incorporate high-quality high-speed circuitry to deliver signals without overshoot or ground bounce for all supported devices. Pin drivers operate down to 1.8V so you'll be ready to program the full range of today's advanced low-voltage devices.

The programmer performs **device insertion test** (wrong device position in socket) and contact check (poor contact pin-to-socket) before it programs each device. These capabilities, supported by signature-byte check help prevent chip damage due to operator error.

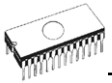
PIKprog+ programmer performs programming **verification** at the marginal level of supply voltage, which, obviously,

improves programming yield, and guarantees long data retention.

PIKprog+ is driven by an easy-to-use control program with pull-down menus, hot keys and on-line help. Selecting of device is performed by its class, by manufacturer or simply by typing a fragment of vendor name and/or part number. Standard device-related commands (read, blank check, program, verify, erase) are enhanced by some test functions (insertion test, signature-byte check), and some special functions (autoincrement). All known data formats are supported. Automatic file format detection and conversion during load of file.

The rich-featured **autoincrement function** enables to assign individual serial numbers to each programmed device - or simply increments a serial number, or the function enables to read serial numbers or any programmed device identification signatures from a file.

Various socket converters are available to handle device in PLCC, SOIC and other packages.



Connecting PIKprog+ programmer to PC

Switch off the PC and programmer. Insert the connection cable, included in the PIKprog+ programmer delivery, to the free printer port of PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counterconnectors. This is very important mainly for the connector to programmer. Though replacing the printer cable by the programmer cable is uncomfortable, it is not recommended to operate the PIKprog+ programmer through a mechanical printer switch. Use of an electronic printer switch isn't possible.

Connect the mains connector of the power supply (or wall-plug power supply self) to a mains plug, connect the connector to the appropriate programmer's connector. Then, on the programmer lights up LED POWER and the programmer PIKprog+ is ready to run. Next switch on the PC and run the control program.

Caution! *If you don't want to switch off your PC when connecting the PIKprog+, proceed as follows*:

- ***When connecting** the programmer to the PC: **FIRST** insert the **communications cable** and **THEN** the **power-supply connector.***
- ***When disconnecting** the programmer from the PC: **FIRST** disconnect the **power-supply connector** and **THEN** the **communication cable.***

Problems related to the PIKprog+ ⇔ PC interconnection, and their removing

If you have any problems with PIKprog+ ⇔ PC interconnection, see section ***Common notes*** please.

In-System serial programming by PIKprog+

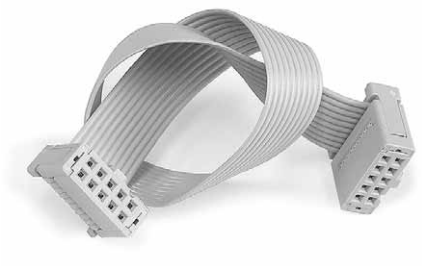
For general definition, recommendation and direction about ISP see section ***Common notes / ISP*** please.

Description of PIKprog+ ISP connector

Pin	Description
2,3,4,6,8	H/L/read
1	VCCP for target device
7,9	GND
10	H/L/read, VPP on MCLR\
5	NC

Specification of ISP connector pins depends on programming device and is displayed at **Additional info window**. These specifications correspond with Microchip application note: In-Circuit Serial Programming™(ICSP™) Guide. Used application note you may find on **www.elnec.com**, section application notes.

Note: *Pin no. 1 is signed by triangle scratch on ISP cable connectors*.



ISP cable of PIKprog+

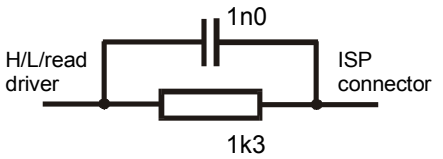
Warnings:

- ***When** you use PIKprog+ as **ISP programmer**, **don't insert** device to **ZIF socket**.*
- ***When** you programm devices in **ZIF socket**, **don't insert** ISP cable to **ISP connector**.*

- *Use only **attached ISP cable**. When you use other ISP cable (other material, length…), programming may occur unreliable.*

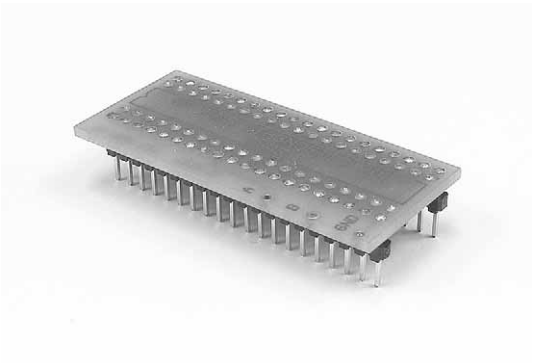
- ***PIKprog+ can supply** programmed device only, but target system **cannot supply PIKprog+**.*
- *PIKprog+ apply programming voltage to target device and checks his value (target system can modify programming voltage). If the programming voltage is different as expected, no action with target device will be executed.*

Note: *H/L/read PIKprog+ driver*



Selftest and calibration

If you have feeling, that your programmer doesn't behave according to yours expectations (but at least every 3 month), please run selftest using Diagnostic POD, that contains standard programmer delivery. For optimal fruit of work with PIKprog+ programmer is recommended every 6 months (no more) to make extended test and check the calibration. All this activities you can find in **Diagnostics** menu of PG4UW.



Manipulation with the programmed device

After selection of desired device for your work, you can insert into the open ZIF socket (the lever is up) and close socket (the lever is down). The correct orientation of the programmed device in ZIF socket is shown on the picture near ZIF socket on the programmer's cover. The programmed device is necessary to insert into the socket also to remove from the socket when LED BUSY light off.

Warning! *PIKprog+ programmer hasn't protection devices, which protect the content of programmed device against critical situations, for example power failures and PC failure (interrupted cable...). Moreover, an device is usually destroyed in the programming mode due to forced interruption of the control program run (Reset or switching the computer off) due to removing the connecting cable, or unplugging the programmed device from the ZIF socket. Incorrectly placed device in the ZIF socket can cause its damage or destruction.*

PIKprog+ specification

Socket, pin drivers and DACs

- FPGA based IEEE 1284 slave printer port, up to 1MB/s transfer rate
- 40-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 40-pins
- two D/A converters for VCCP and VPP, controllable rise and fall time
- VCCP range 0..7V/250mA
- VPP range 0..15V/150mA
- pindriver: 40 TTL pindrivers, special GND/VCC/VPP pindriver for PICmicro® devices
- spare GND, VCC and VPP driver, which add additional made-by-wire GND/VCC/VPP pin capability for future devices
- FPGA based TTL driver provides H, L, CLK, pull-up, pull-down on all pindriver pins
- level H selectable from 1.8 V up to 5V
- in-circuit serial programming (ISP) capability included
- continuity test: each pin is tested before every programming operation
- selftest capability
- autocalibration

Device support

- microcontrollers MicrochipTM PICmicro®: 12xxx, 14xxx, 16xxx, 17xxx and 18xxx series, 8 to 40 pin (*1), parallel and serial mode
- Serial E(E)PROM: 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx series

  *(*1) - suitable adapters are available for non-DIL packages and devices with more than 40 pins*

Programming speed

Note. *These times strongly depend on PC speed, LPT port type and operating system free resources. Therefore are given values of two different PC configurations for comparison.*

Device	Operation	Time A	Time B
PIC16C67	programming and verify	13 sec	11 sec
PIC18F452	Programming and verify	11 sec	9 sec

Time A conditions: *Pentium MMX, 250 MHz, ECP/EPP, WIN98.*

Time B conditions: *Athlon, 750 MHz, ECP/EPP on PCI bus, WIN98.*

Device operations

- **standard:**
  - intelligent device selection by device type, manufacturer or typed fragment of part name
  - blank check, read, verify
  - program
  - erase
  - configuration and security bit program
  - illegal bit test
  - checksum
- **security**
  - insertion test
  - contact check
  - ID byte check
- **special**
  - auto device serial number increment
  - statistic
  - count-down mode

Buffer operations

- view/edit, find/replace
- fill, copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

File load/save

- no download time because programmer is PC controlled
- automatic file type identification

Supported file formats

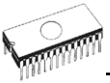
- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S-record, MOS, Exormax, Tektronix, ASCII-SPACE-HEX

PC system requirements

See section ***Introduction/ PC requirements***

General:

- operating voltage 12..15V DC/max. 0.5A (adapter included in package)
- power consumption max. 5W active, about 0.5W sleep
- dimensions 132x66x30 [mm] (5.2x2.6x1.2 [inch])
- weight (without external power adapter) ca. 200g
- temperature 5°..40°C
- humidity 20%..80%, non condensing



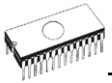
Package included

- PIKprog+ programmer
- connection cable PC-programmer
- ISP cable
- diagnostic POD for selftest
- anti-dust cover for ZIF socket
- wall plug adapter 220V to 12V DC/500mA, unstabilized
- user manual
- software
- "DEVICE PROBLEM REPORT" form
- registration card
- transport case

Additional services

- Keep Current
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.

PIKprog



Introduction

PIKprog is little and powerful portable programmer for Microchip PIC series of microcontrollers. PIKprog enables also programming serial EEPROM with interface types IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx). The programmer is equipped by DIL ZIF 40pin socket. The quality of programmer is completed by comfortable control program.

PIKprog interfaces with the IBM PC, AT or higher, portable or desktop personal computers. Programmer allows you to directly connect to your PC through any standard parallel (printer) port - no special interface card is needed.

The advantage of PIKprog is its easy-to-use comfortable software with pull-down menu, hot keys and on-line help. You can select device by classes, by manufacturers or simply type device vendor's name and part number. The standard device operation functions (read, blank check, program, verify) have been completed with some test functions. The program facilitates the use of the buffer and files, including automatic file format detection and conversion.

For PIKprog programmer are available DIL to SOIC socket converters.

Connecting PIKprog programmer to PC

Switch off the PC and programmer. Insert the connection cable, included in the PIKprog programmer delivery, to the free printer port of PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counterconnectors. This is very important mainly for the connector to programmer. Though replacing the printer cable by the programmer cable is uncomfortable, it is not recommended to operate the PIKprog programmer through a mechanical printer switch. Use of an electronic printer switch isn't possible.

Connect the mains connector of the power supply (or wall-plug power supply self) to a mains plug, connect the connector to the appropriate programmer's connector. Then, on the programmer lights up LED POWER and the programmer PIKprog is ready to run. Next switch on the PC and run the control program.

Caution! *If you don't want to switch off your PC when connecting the PIKprog, proceed as follows:*

- ***When connecting** the programmer to the PC: **FIRST** insert the **communications cable** and **THEN** the **power-supply connector.***
- ***When disconnecting** the programmer from the PC: **FIRST** disconnect the **power-supply connector** and **THEN** the **communication cable**.*

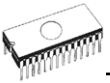
Problems related to the PIKprog ⇔ PC interconnection, and their removing

If you have any problems with PIKprog ⇔ PC interconnection, see section ***Common notes*** please.

Manipulation with the programmed device

After selection of desired device for your work, you can insert into the open ZIF socket (the lever is up) and close socket (the lever is down). The correct orientation of the programmed device in ZIF socket is shown on the picture near ZIF socket on the programmer's cover. The programmed device is necessary to insert into the socket also to remove from the socket when LED BUSY light off.

Warning! *PIKprog programmer hasn't protection devices, which protect the content of programmed device against critical situations, for example power failures and PC failure*



(interrupted cable...). Moreover, an device is usually destroyed in the programming mode due to forced interruption of the control program run (Reset or switching the computer off) due to removing the connecting cable, or unplugging the programmed device from the ZIF socket. Incorrectly placed device in the ZIF socket can cause its damage or destruction.

PIKprog specification

Socket, pin drivers and DACs

- 40-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 40-pin
- TTL driver provides H, L and read for all pins

Device support

- microcontrollers Microchip PIC series: 12xxx, 14xxx, 16xxx, 8/16/18/28/40 pins
- Serial E(E)PROM: 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx series

Note:
Serial EEPROMs programming is performed little bit slowly.

Device operations

- **standard:**
  - blank check
  - read
  - program
  - verify
  - erase
  - configuration and security bit program
- **security:**
  - ID byte check
- **special:**
  - auto device serial number increment

Buffer operations

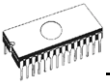
- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Supported file formats

- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-space-HEX, Tektronix

PC system requirements

See section ***Introduction/ PC requirements***



General

- operating voltage 15÷20V DC/max. 200mA
- power consumption 3W max.
- dimensions 132x66x30 [mm] (5.2x2.6x1.2 [inch])
- weight (without external adapter) ca. 200g
- temperature 5° ÷ 40°C
- humidity 20%..80%, non condensing

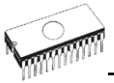
Package included

- PIKprog programmer
- 1.3m (4.3 ft.) connecting cable PIKprog to (IBM) PC
- wall plug adapter 15V DC/ 500mA, unstabilized
- software
- user manual
- “DEVICE PROBLEM REPORT" form
- registration card
- transport packing

Additional services

- Keep Current
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.

SEEprog



Introduction

SEEprog is universal programmer of all serial EEPROM in 8-pin DIL package. SEEprog programs EEPROM with interface IIC, SPI and Microwire, and also specialty as for example digital thermometers. The programmer supports LV (3.3V) devices too.

SEEprog interfaces with the IBM PC, AT or higher, portable or desktop personal computers. Programmer allows you to directly connect to your PC through any standard parallel (printer) port - no special interface card is needed.

The advantage of SEEprog is its easy-to-use comfortable software with pull-down menu, hot keys and on-line help. You can select device by classes, by manufacturers or simply type device vendor's name and part number. The standard device operation functions (read, blank check, program, verify) have been completed with some test functions. The program facilitates the use of the buffer and files, including automatic file format detection and conversion.

For SEEprog programmer are available DIL to SOIC socket converters.

Connecting SEEprog programmer to PC

Switch off the PC and programmer. Insert the connection cable, included in the SEEprog programmer delivery, to the free printer port of PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counterconnectors. This is very important mainly for the connector to programmer. Though replacing the printer cable by the programmer cable is uncomfortable, it is not recommended to operate the SEEprog programmer through a mechanical printer switch. Use of an electronic printer switch isn't possible.

Connect the mains connector of the power supply (or wall-plug power supply self) to a mains plug, connect the connector to the appropriate programmer's connector. Then, on the programmer lights up LED POWER and the programmer SEEprog is ready to run. Next switch on the PC and run the control program.

Caution! *If you don't want to switch off your PC when connecting the SEEprog, proceed as follows:*

- ***When connecting** the programmer to the PC: **FIRST** insert the **communications cable** and **THEN** the **power-supply connector.***
- ***When disconnecting** the programmer from the PC: **FIRST** disconnect the **power-supply connector** and **THEN** the **communication cable**.*

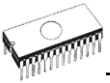
Problems related to the SEEprog ⇔ PC interconnection, and their removing

If you have any problems with SEEprog ⇔ PC interconnection, see section ***Common notes*** please.

Manipulation with the programmed device

After selection of desired device for your work, you can insert into the open ZIF socket (the lever is up) and close socket (the lever is down). The correct orientation of the programmed device in ZIF socket is shown on the picture near ZIF socket on the programmer's. The programmed device is necessary to insert into the socket also to remove from the socket when LED BUSY light off.

Warning! *SEEprog programmer hasn't protection devices, which protect the content of programmed device against critical*



situations, for example power failures and PC failure (interrupted cable...). Moreover, an device is usually destroyed in the programming mode due to forced interruption of the control program run (Reset or switching the computer off) due to removing the connecting cable, or unplugging the programmed device from the ZIF socket. Incorrectly placed device in the ZIF socket can cause its damage or destruction.

SEEprog specifications

Socket and control of pins:

- DIL/ZIF socket (300mil)
- each pin is possible to set in position Low and Pull-up
- from each pin is possible to read
- support of Low Voltage devices

For works with serial EEPROM are used nearest 8 pins to lever, unused pins are not connected.

Supported devices:

- EEPROM IIC (24Cxxx)
- EEPROM Microwire (93Cxxx)
- EEPROM SPI (25Cxxx)
- specially devices (digital thermometers, ...)

Device operations:

- **standard**:
  - blank check
  - read
  - program
  - verify
  - checksum

Buffer operations

- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

File format conversion

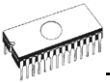
- binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, ASCII - space - HEX

PC systems requirements

See section ***Introduction/ PC requirements***

General:

- power supply: 8V...20V/DC, max.50mA
- dimensions: 132x66x30 [mm]
- mass: ca 150g
- temperature range: 5..40°C



Package included

- SEEprog programmer
- 1.3m (4.3 ft.) connecting cable SEEprog to (IBM) PC
- wall plug adapter 12V DC/ 500mA, unstabilized
- software
- user manual
- "DEVICE PROBLEM REPORT" form
- registration card
- transport packing

Additional services

- Keep Current
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.

Software

The programmer software

The programmer package contains a CD with the control program, useful utilities and additional information. The permission to freely copy the content of the CD is granted in order to demonstrate how ELNEC's programmers work. Differences and modifications to this manual (if they exist) may be found in **www.elnec.com** web site.

Installing of programmer software

Installing the programmer software is very easy. Insert delivered CD to your CD drive and install program starts automatically. Install program (setup.exe), which will guide you through the installation process and which will do all the necessary steps before you can first run the control program.

If you use the operating system **DOS**, run **Install** program from CD.

Programs PG4U.EXE (for DOS) and PG4UW.EXE (for Windows) are common control programs for all ELNEC's programmers. We guarantee running of these programs under all of above mentioned operating systems without any problems. Also background operation under Windows is error-free.

For DOS versions of the control program running under WINDOWS 3.11/95/98, we suggest the following: (adjust "Properties" on the icon on your screen):

- Run the control program in **full-screen DOS** (in text mode). In graphic mode the control program might be somewhat slower.
- If you want the control program to run in the **background**, it is necessary to set: (WIN95/98) RIGHT mouse button on the DOS icon/ Properties/ Others/ Background/ Enable; (WIN3xx) in the appropriate PIF file, set the option to Background

New versions of programmer software

In order to exploit all the capabilities of programmer we recommend using the latest version of PG4U/PG4UW (see appendix B - Keep-Current Service). You may download the latest version of programmer software (file PG4UARC.EXE /PG4UWARC.EXE) from our Internet site **www.elnec.com**, part download. You may also obtain CD with this file by snail-mail (a mailing charge will apply).

Upgrading the programmer software

Copy PG4UARC.EXE / PG4UWARC.EXE to a temporary directory then launch it. After extraction you will see all available files needed for the installation process. Then redo a standard installation (run the Setup / Install program). You may delete all files from the temporary folder after the installation process is complete.

Using the programmer software

The control program delivered by ELNEC, included on the CD in your package, is granted to be free from any viruses at the moment of delivery. To increase their safety our programs include a special algorithm for detecting possible virus infections.

Run the control program

From command line of MS/PC-DOS write:

PG4U

in Windows environment: double click to icon PG4UW.

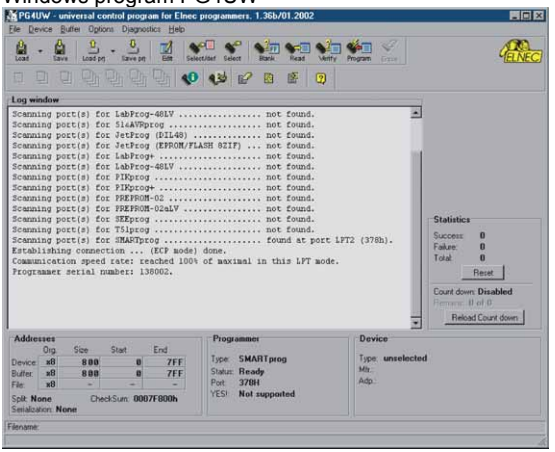
After start, control program PG4U / PG4UW automatically scan all existing ports and search for the connected any ELNEC's programmer. Program PG4U/PG4UW is common for all the ELNEC's programmers, hence program try to find all supported (JetProg, LabProg+, SmartProg, PREPROM-02aLV, T51prog, 51&AVRprog, PIKprog+, PIKprog and SEEprog) programmers.

Notes: *When the PG4U/PG4UW program is started, program is checked for its integrity. Than the program display a standard user menu and waits for your instructions.*

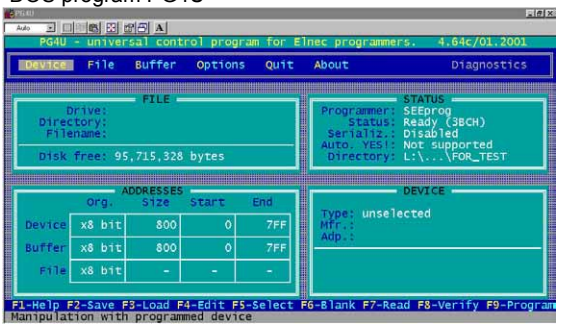
If the control program cannot communicate with the programmer, an error message appears on the screen, including error code and description of possible reasons (disconnected programmer, bad connection, power supply failure, incompatible printer port, ...). Eliminate the error source and press any key. If error condition still exists, the program resumes its operation in the demo mode and access to the programmer is not possible. If you cannot find the cause of the error, follow the instructions in **Troubleshooting** section. In addition, the control program checks communication with programmer prior to any operation with the programmed device.

Description of the user screen

Windows program PG4UW


PG4UW - universal control program for Elnec programmers. 1.36b/01.2002
File Device Buffer Options Diagnostics Help
Log window
Scanning port(s) for LabProg-48LV ..... not found.
Scanning port(s) for SLeAVRprog ..... not found.
Scanning port(s) for JetProg (DIL48) ..... not found.
Scanning port(s) for JetProg (EPROM/FLASH 8BIF) ... not found.
Scanning port(s) for LabProg+ ..... not found.
Scanning port(s) for LabProg-48LV ..... not found.
Scanning port(s) for PIKprog ..... not found.
Scanning port(s) for PIKprog+ ..... not found.
Scanning port(s) for PREPROM-02 ..... not found.
Scanning port(s) for PREPROM-02aLV ..... not found.
Scanning port(s) for SEEprog ..... not found.
Scanning port(s) for T51prog ..... not found.
Scanning port(s) for SMARTprog ..... found at port LPT2 (378h).
Establishing connection ... (ECP mode) done.
Communication speed rate: reached 100% of maximal in this LPT mode.
Programmer serial number: 136002.
Statistics
Success: 0
Failure: 0
Total: 0
Reset
Count down: Disabled
Reload Count down
Addresses
Org. Size Start End
Device: x8 800 0 7FF
Buffer: x8 800 0 7FF
File: x8 - - -
Split: None
Serialization: None
CheckSum: 0007F800h
Programmer
Type: SMARTprog
Status: Ready
Port: 378H
YES!: Not supported
Device
Type: unselected
Mfr.:
Adp.:
Filename:

DOS program PG4U



Header bar	the name, copyright statement and version of the PG4U/PG4UW the control program
Menu bar	list of basic functions
File window/**Filename**	information on the currently loaded file in buffer
Status window	information about the status of the programmer and PG4U/PG4UW
Addresses window	organization, size, start and end addresses of the target device, buffer and file
Device window	all relevant information about the current target device
Help bar	a brief description of selected command

Menu selection is carried out in the normal GUI fashion - either by cursor moving plus pressing **<Enter>**, or by typing the highlighted letter in the wanted menu or - of course - by mouse. Hot-keys are available for even quicker selection of intensely used commands.

Note: *Data entered through keyboard is in HEX format, excepting ASCII blocks in Buffer/View/Edit command*.

List of hot keys

<F1>	Help	Calls Help
<F2>	Save	Save file
<F3>	Load	Load a file into the buffer
<F4>	Edit	Viewing/editing of buffer
<F5>	Select/default from	Target-device selection 10 last selected devices list
<Alt+F5>	Select/manual by	Target-device selection typing device/vendor name
<F6>	Blank	Blank check
<F7>	Read	Reads device's content into the buffer
<F8>	Verify	Compares contents of the target device with the buffer
<F9>	Program	Programs target device
<Alt+Q>	Exit without save	Terminates the PG4U / PG4UW
<Alt+X>	Exit and save	Terminates the PG4U / PG4UW and saving settings too
<Ctrl+F1>		Displays additional information about current device
<Ctrl+F2>	Erase	Fill's the buffer with a given value
<Ctrl+Shift+F2>		Fill's the buffer with random values.

File

This submenu is used for source files manipulation, settings and viewing directory, changes drives, changes start and finish address of buffer for loading and saving files by **binary**, **MOTOROLA**, **MOS Technology**, **Intel (extended) HEX**, **Tektronix, ASCII space**, **JEDEC**, and **POF** format.

File / Load

Analyse file format and loads the data from specified file to the buffer. You can choose the format desired (**binary**, **MOTOROLA**, **MOS Technology**, **Tektronix**, **Intel (extended) HEX**, **ASCII space**, **JEDEC** and **POF**). The control program stores a last valid mask for file listing. You can save the mask into the config. file by command **Options / Save options**.

Checking the check box **Automatic file format recognition** tells program to detect file format automatically. When program can't detect file format from one of supported formats, the binary file format is assumed.

When the check box **Automatic file format recognition** is unchecked program allows user to manually select wished file format from list of available file formats on panel **Selected file format**. When Binary file format is selected, there can be specified Buffer start value. Buffer start value is buffer address from which data read from file will be written to buffer.

The reserved key **<F3>** will bring out this menu from any menu and any time.

File / Save

Saves data in the buffer, which has been created, modified, or read from a device onto a specified disk. You can choose the format desired (**binary**, **MOTOROLA**, **MOS Technology**, **Tektronix**, **Intel (extended) HEX**, **ASCII space**, **JEDEC** and **POF**).

The reserved key **<F2>** will bring out this menu from any menu and any time.

File / Load project

This option is used for loading project file, which contains device configuration buffer data saved and user interface configuration.

The standard dialog **Load project** contains additional window - **Project description** - placed at the bottom of dialog. This

window is for displaying information about currently selected project file in dialog Load project.
Project information consists of:

- name and manufacturer of the first device selected in the project
- date and time of project creation
- version of program by which project was created
- user written description of project (it can be arbitrary text, usually author of project and some notes)

File / Save project

This option is used for saving project file, which contains settings of device configuration and buffer data saved. Data saved to project file can be restored anytime by menu command **File / Load project**.

The dialog **Save project** contains three additional windows in **Project description** panel placed at the bottom of dialog **Save project**. The windows are for displaying information about currently selected project file in dialog **Save project** and information about current project, which has to be saved.

Project information consists of:

- name and manufacturer of the first device selected in the project
- date and time of project creation
- version of program by which project was created
- user written description of project (it can be arbitrary text, usually author of project and some notes)

The first (upper) window contains information about currently selected project file in dialog Save project.

The second (middle) windows displays information about actual program configuration including currently selected device, active programmer, date, time and program version. These actual program settings are used for creation of project description header.

The third (bottom) window is user editable and contains project description (arbitrary text), which usually consists of project author and some notes.

File / Reload file

Choose this option to reload a recently used file.

When you use a file, it is added to the **Reload file** list. Files are listed in order depending on time of use of them. Lastly used files are listed before files used far off.

To Reload a file:

1. From the File menu, choose Reload file.
2. List of lastly used files is displayed. Click the file you want to reload.

Note: *When reloading a file the file format is used, by which the file was lastly loaded/saved.*

File / Reload project

Choose this option to reload a recently used project.

When you use a project, it is added to the **Reload project** list. Projects are listed in order depending on time of use of them. Lastly used projects are listed before projects used far off.

To Reload a project:

1. From the File menu, choose Reload project.
2. List of lastly used projects is displayed. Click the project you want to reload.

File / Project options

This option is used for display/edit project options of actually loaded project. Project options means basic description of project including following project data:

- device name and manufacturer
- project creation date
- version of program by which project was created
- user defined project description (arbitrary text), e.g. project author and other text data for more detailed project description

User can directly edit user defined project description only. Device name, manufacturer, project date and program version are generated automatically by program.

File / Load encryption table

This command loads the data from binary file from disk and it saves them into the part of memory, reserved for an encryption (security) table.

File / Save encryption table

This command writes the content of the memory's part, reserved for an encryption table, into the file on the disk as a binary data.

File / Exit without save

The command deallocates heap, cancels buffer on disk (if exists) and returns back to the operation system.

File / Exit and save

The command deallocates heap, cancels buffer on the disk (if exists), saves current setting of last 10 selected devices to disk and returns back to the operation system.

Device

The functions for a work with selected programmable devices - device select, read data from device, device blank check, device program, device verify and device erase.

Device / Select from default devices

This window allows selecting the desired type of the device from list of default devices. This one is a cyclic buffer in which are stored last 10 selected devices including its device options. This list is saved to disk by command **File / Exit and save**.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) that supported this device. You can find here a package information and other general information about current device too.

Use a **<Del>** key for delete of current device from list of default devices. There isn't possible to empty this list, if you repeat this access. The last device stays in buffer and the **<Del>** key isn't accepted.

Device / Select device ...

This window allows selecting the desired type of the device from all devices supported by current programmer. It is possible to choose device by **name**, by **type** or by **manufacturer**.

Selected device is automatically saved to buffer of default devices (max. 10 devices). This buffer is accessible with **Device / Select from default devices** command.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) that supported this device. You can find here a package information and other general information about current device too.

Select device ... / All

This window allows selecting the desired type of the device from all devices supported by current programmer. Supported devices are displayed in a list box.

Device can be select by double click on a line from list with desired manufacturer name and device number or by entering manufacturer name and/or device number in a search box (use

a key **<Space>** as a separation character) and press **<Enter>** or click **OK** button.

Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

Selected device is automatically saved to buffer of default devices (max. 10 devices). This buffer is accessible with **Device / Select from default devices** command.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules), which supported this device. You can find here a package information and other general information about current device too.

Select device ... / Only selected type

This window allows selecting the desired type of the device. At the first - you must select a device type (e.g. EPROM) and device subtype (e.g. 64Kx8 (27512)), using mouse or cursor keys. It will cause a list of manufacturers and devices will be displayed.

Device can be select by double click on a line from list with desired manufacturer name and device number or by entering manufacturer name and/or device number in a search box (use a key **<Space>** as a separation character) and press **<Enter>** or click **OK** button.

Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

Selected device is automatically saved to buffer of default devices (max. 10 devices). This buffer is accessible with **Device / Select from default devices** command.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) that supported this device. You can find here a package information and other general information about current device too.

Select device ... / Only selected manufacturer

This window allows selecting the desired device type by manufacturer. First select a required manufacturer in Manufacturer box using mouse or cursor keys. It will cause a list of selected manufacturer devices will be displayed.

Device can be select by double click on a line from list with desired manufacturer name and device number or by entering

device number in a search box (use a key **<Space>** as a separation character) and press **<Enter>** or click **OK** button.

Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

Selected device is automatically saved to buffer of default devices (max. 10 devices). This buffer is accessible with **Device / Select from default devices** command.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) that supported this device. You can find here a package information and other general information about current device too.

Device / Select EPROM /Flash by ID

Use this command for autoselect an EPROM or Flash as active device by reading the device ID. The programmer can automatically identify certain devices by the reading the manufacturer and the device-ID that are burnt into the chip. This only applies to EPROM or Flash that supports this feature. If the device does not support a chip ID and manufacturer's ID, a message will be displayed indicating this as an unknown or not supported device.

If more devices with identical chip ID and manufacturer's ID were detected, the list of these devices will be displayed. A corresponding device can be chosen from this list by selecting its number (or manufacturer name) from list and press **<Enter>** (or click **OK** button). Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

WARNING: *The control program only support this time EPROM's and Flash with 28 and 32 pins. Any of programmers determines pins number automatically. For other programmers you must enter this number manually.*

The programmer applies a high voltage to the appropriate pins on the socket. This is necessary to enable the system to read the device ID. Do not insert into the socket a device that is not an EPROM or Flash. It may be damaged when the programmer applies the high voltage.

We don't recommend apply this command to 2764 and 27128 EPROM types, because most of them ID not supports.

Device / Device options

All settings of this menu are used for programming process, serialization and associated file control.

Device / Device options / Operation options

All settings of this command are used for programming process control. This is a flexible environment, which content items associated with current device and programmer type. Items, which are valid for the current device but aren't supported by current programmer, are disabled. These settings are saving to disk along with associated device by **File / Exit and save** command.

List of items:

- group **Addresses**:

device start address	(default 0)
device end address	(default device size-1)
buffer start address	(default 0)

- group **Insertion test:**

insertion test	(default ENABLE)
check ID bytes	(default ENABLE)

- group **Command execution:**

blank check before programming	(default DISABLE)
erase before programming	(default DISABLE)
verify after reading	(default ENABLE)
verify after programming	(ONCE, TWICE)
verify options	(nominal VCC 5%, nominal VCC 10%, VCCmin VCCmax)

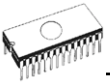
Device / Device options / Serialization

Serialization is special mode of program. When a serialization mode is activated, a specified value is automatically inserted on predefined address into buffer before programming each device. When more devices are programmed one by one, the serial number value is changed for each device automatically and inserted into buffer before programming device, so each device has unique serial number.

There are two types of serialization:

- Incremental mode
- From file mode

If a new device is selected, the serialization function is set to a default state i.e. disabled.



Actual serialization settings for actually selected device are saving to disk along with associated device by **File / Exit and save** command.
When incremental mode is active following actual settings are saved to configuration file: address, size, serial value, incremental step and settings of modes ASCII / BIN, DEC / HEX, LS byte / MS Byte first.

When from-file mode is active following actual settings are saved to configuration file: name of input serialization file and actual label, which indicates the line with actual serial number in input file.

When program is in multiprogramming mode (multiple socket programmer is actually selected) the special section - **Action on not programmed serial values due to error** - is displayed in dialog **Serialization**. In this section two choices are available:

1. Ignore not programmed serial values
2. Add not programmed serial values to file

Ignore not programmed serial values means the not programmed serial values are ignored and no action is done with them.

Add not programmed serial values to file means the not programmed serial values are added to file. The file of not programmed serial values has the same text format as serialization file for ”From-file” serialization mode. So there is possible to program the serial values later on by ”From-file” serialization mode.

If device programming is stopped by user, program will not change the serial values ready for next batch of devices. The same situation is if device program is incomplete, e.g. for device insertion test error.

Ignoring or writing not programmed serial values is only used when at least one device from current batch of devices in multiple socket module programmer is completely programmed and verified without errors.

Device / Device options / Serialization / Incremental mode

The **Incremental mode** enables to assign individual serial numbers to each programmed device. A starting number entered by user will be incremented by specified step for each device program operation and loaded in selected format to specified buffer address prior to programming of each device.

There are following options, that user can modify for incremental mode:

S / N size
S / N size option defines the number of bytes of serial value which will be written to buffer. For Bin (binary) serialization modes values 1-4 are valid for S / N size and for ASCII serialization modes values 1-8 are valid for S / N size.

Address
Address option specifies the buffer address, where serial value has to be written. Note that address range must be inside the device start and device end addresses. Address must be correctly specified so the last (highest or lowest) byte of serial value must be inside device start and device end address range.

Start value
Start value option specifies the initial value, from which serialization will start. Generally the max. value for serialization is $1FFFFFFF in 32 bit long word.
When the actual serial value exceeds maximum value, three most significant bits of serial number are set to zero. After this action the number is always inside 0..$1FFFFFFF interval (this is basic style of overflow handling).

Step
Step options specifies the increment step of serial value incrementation.

S / N mode
S / N mode option defines the form in which serial value has to be written to buffer. Two options are available:

- ASCII
- Bin

ASCII - means the serial number is written to buffer as ASCII string. For example number $0528CD is in ASCII mode written to buffer as 30h 35h 32h 38h 43h 44h (‘0’ ‘5’ ‘2’ ‘8’ ‘C’ ‘D’), i.e. six bytes.
Bin - means the serial number is written directly to buffer. If the serial number has more than one byte length, it can be written in one of two possible byte orders. The byte order can be changed in „Save to buffer“ item.

Style
Style option defines serial number base. There are two options:

- Decimal
- Hexadecimal.

Decimal numbers are entered and displayed using the characters '0' through '9'.
Hexadecimal numbers also use characters 'A' through 'F'.
The special case is Binary Dec, which means BCD number style. BCD means the decimal number is stored in hexadecimal number, i.e. each nibble must have value from 0 to 9. Values A to F are not allowed as nibbles of BCD numbers.
Select the base in „Style“ options before entering numbers of serial start value and step.

Save to buffer
Save to buffer option specifies the serial value byte order to write to buffer. This option is used for Bin S / N mode (for ASCII mode it has no effect).

Two options are available:

- LSByte first (used by Intel processors) will place the Least Significant Byte of serial number to the lowest address in buffer.
- MSByte first (used by Motorola processors) will place the Most significant Byte first to the lowest address in buffer.

Device / Device options / Serialization / From file mode

Using the From-file method, serial values are read from the user specified input file and written to buffer on address specified in input file.
There are two user options: File name and Start label.

File name
File name option specifies the file name from which serial addresses and values will be read. The input file for From file serialization must have special format, which is described in From file serialization file format below.

Start label
Start label defines the start label in input file. The reading of serial values from file starts from defined start label.

From file serialization file format
From file serialization input file includes addresses and arrays of bytes defining buffer addresses and data to write to buffer. Input file has text type format, which structure is:

[label*1*] addr byte*0* byte*1* .. byte*n*
...
[label*n*] addr byte*0* byte*1* .. byte*m* , addr byte*0* byte*1* ... byte*k*
_____/ _____/
| |
basic part optional part

; Comment

meaning is:

basic part
Basic part defines buffer address and array of bytes to write to buffer. Basic part must be always defined after label in line.

optional part
Optional part defines the second array of bytes and buffer address to write to buffer. One optional part can be defined after basic part of data.

label*1*, label*n* - labels
Labels are identifiers for each line of input file. They are used for addressing each line of file. The labels should be unique. Addressing lines of file means, the required start label entered by user defines line in input file from which serial values reading starts.

addr -
Addr defines buffer address to write data following the address.

byte*0*..byte*n*, byte*0*..byte*m*, byte*0*..byte*k* -
Bytes arrays byte*0*..byte*n*, byte*0*..byte*m* and byte*0*..byte*k* are defining data, which are assigned to write to buffer. Maximum count of bytes in one data field following the address is 64 bytes. Data bytes are written to buffer from address addr to addr+n.
The process of writing particular bytes to buffer is:
byte*0* to addr
byte*1* to addr + *1*
byte*2* to addr + *2*
....
byte*n* to addr + *n*

Optional part is delimited from the first data part by character " , " (comma) and its structure is the same as in the first data part, i.e. address and following array of data bytes.

Characters with special use:

[ ] - labels must be defined inside square brackets

',' – character which delimiters basic part and optional part of data

‘;’ - the semicolon character means the beginning of a comment. All characters from „;„ to the end of line are ignored. Comment can be on individual line or in the end of definition line.

Note:

- *Label names can contain all characters except ‘[‘ and ‘]’. The label names are analysed as non case sensitive, i.e. character ‘a’ is same as ‘A’, ‘b’ is same as ‘B’ etc..*
- *All address and byte number values in input file are hexadecimal.*
- *Allowed address value size is from 1 to 4 bytes.*
- *Allowed size of data arrays in one line is in range from 1 to 64 bytes. When there are two data arrays in one line, the sum of their size in bytes can be maximally 80 bytes.*
- *Be careful to set correct addresses. Address must be defined inside device start and device end address range. In case of address out of range, warning window appears and serialization is set to disabled (None).*

Example:

[nav*1*] A7890 78 89 56 02 AB CD ; comment1
[nav*2*] A7890 02 02 04 06 08 0A
[nav*3*] A7890 08 09 0A 0B A0 C0 ; comment2
[nav*4*] A7890 68 87 50 02 0B 8D
[nav*5*] A7890 A8 88 59 02 AB 7D

;next line contains also second definition
[nav*6*] A7890 18 29 36 42 5B 6D , FFFF6 44 11 22 33 99 88 77 66 55 16

; this is last line - end of file

In the example file six serial values with labels „nav*1*“, „nav*2*“, ...“nav*6*“ are defined. Each value is written to buffer on address $A7890. All values have size 6 bytes. The line with „nav*6*“ label has also second value definition, which is written to buffer on address $FFFF6 and has size 10 bytes, i.e. the last byte of this value will be written to address $FFFFF.

Device / Device options / Statistics

Statistics gives the information about actual count of device operations, which were proceeded on selected type device. If one device is corresponding to one device operation, e.g. programming, the number of device operations will be equal to number of programmed devices.

The next function of statistics is **Count down**. Count down allows checking the number of device operations, and then number of devices, on which device operations have to be done. After each successful device operation the value of

count down counter is decremented. Count down has user defined start number of devices to do. When count down value reach zero, it means, specified number of devices is complete and user message about complete count down will be displayed.

Statistics dialog contains following options:

Check boxes **Program**, **Verify**, **Blank**, **Erase** and **Read** define operations, after which statistics values increment.

Check box **Count down** sets Count down activity (enable or disable). Edit box following the Count down check box defines initial number of count down counter, from which count down starts.

Statistics dialog can be also opened by pressing right mouse button on Statistics panel and clicking displayed item Statistics.

Actual statistics values are displaying in main window of control program in Statistics panel.

Statistics panel contains three statistics values – **Success**, **Failure**, **Total** and two **Count down** information values **Count down** and **Remains**.

Meaning of the values is:

Success number of operations which where successfully completed

Failure number of operations which where not successfully completed

Total number of all operations

Count down informs about Count down activity (Enabled or Disabled)

Remains informs about remaining number of device operations to do

Successful operation means any device operation of these, which is completed without errors:

- program
- verify
- blank check
- erase
- read

If device operation is finished with error(s) it is not successful operation.

When new device type is selected, all statistics values are set to zero and **Count down** is set to **Disabled**.

Reset button in **Statistics** panel reset statistics values.
Reload Count down button in **Statistics** panel reloads initial value to **Count down**.

Device / Device options / Associated file

This command is used for setting associated file with current device. This is a file, which can be automatic loaded to buffer after device is selected from default devices select list or by start control program.

You can edit the associated file name in file name box, put a full pathname. The control program checks the present of this file on the disk. Also is possible enabling or disabling automatic load of this file.

You can save both settings i.e. associated file and enabling of automatic load of this file to disk by command **File / Exit and save**.

Device / Blank check

This command allows to blank check of all devices or its part if possible. The control program reports a result of this action by a write of a warning message to INFO window.

The menu command **Device / Device options / Operation options** allows to set another working area as the standard.

Device / Read

This command allows to read all device or its part into the buffer. The control program reports a finish of this action by write a message to INFO window.

The menu command **Device / Device options / Operation options** allows to set another working area as the standard. Setting an option Verify data after reading in this menu command means a higher reliability for device reading.

Device / Verify

This command compares the programmed data of the all device or its part with data in buffer. The control program reports a result of this action by a write of an error message to INFO window.

The menu command **Device / Device options / Operation options** allows to set another working area as the standard.

By the setting in the menu **Options / Display errors** the command lets to write the found errors on the display or write the found errors to VERIFY.ERR file. In the Display errors mode to the screen can display the program max. 45 the first

found differences, which are located by the address where they were caused.

Device / Program

This command allows to programming of the all device or its part by the data of the buffer. The control program reports a result of this action by a write of an error message to INFO window.

The menu command **Device / Device options / Operation options** allows to set another working area as the standard, and set other operation options for programming process control.

Device / Erase

This command allows to erase the all programmable device. The program reports the end without error or end with the error by writes the warning report on the display.

Device / Test

This command executes a test with device selected from list of supported devices (e.g. static RAM) on programmers, which support this test.

Device / Device info

The command provides additional information about the current device - size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) that supported this device. You can find here a package information and other general information about current device too.

The reserved key **<Ctrl+F1>** will bring out this menu from any menu and any time immediately.

Buffer

This submenu is used for buffer manipulation, block operation, filling a part of buffer with string, erasing, checksum and of course editing and viewing with other items (find and replace string, printing...).

Buffer / View/Edit

This command is used to view (view mode) or edit (edit mode) data in buffer (for viewing in DUMP mode only). Use arrow keys for select the object for edit. Edited data are signified by colour.

You can use <F4> hot key also.

View/Edit Buffer

F1	display help of actual window
F2	fill block causes filling selected block of buffer by requested hex (or ASCII) string. Sets start and end block for filling and requested hex or ASCII string.
Ctrl+F2	erase buffer with specified blank value
Ctrl+Shift+F2	fill buffer with random data
F3	copy block is used to copy specified block of data in current buffer on new address. Target address needn't be out from source block addresses.
F4	move block is used to move specified block of data in current buffer on new address. Target address needn't be out from source block addresses. Source address block (or part) will be filled by topical blank character.
F5	swap bytes command swaps a high- and low- order of byte pairs in current buffer block. This block must started on even address and must have an even number of bytes. If this conditions do not fulfil, the program modifies addresses itself (start address is moved on lower even address and/or end address is moved on higher odd address).
F6	print buffer
F7	find string (max. length 16 ASCII characters)
F8	find and replace string (max. 16 ASCII chars.)
F9	change current address
F10	change mode view / edit

F11	switch the mode of buffer data view between 8 bit and 16 bit view. It can be also do by mouse clicking on the button to the right of View/Edit mode buffer indicator. This button indicates actual data view mode (8 bit or 16 bit), too.
F12	checksum dialog allows to count checksum of selected block of buffer change mode view / edit
Arrow keys	move cursor up, down, right and left
Home/End	jump on start / end current line
PgUp/PgDn	jump on previous / next page
Ctrl+PgUp/PgDn	jump on start / end current page
Ctrl+Home/End	jump on start / end current device
Backspace	move cursor one position left (back)

Note: *characters 20H - FFH (mode ASCII) and numbers 0..9, A..F (mode HEX) immediately changes content of edit area.*

Print buffer

This command allows write selected part of buffer to printer or to file. Program uses at it an external text editor in which selected block of buffer is displayed and can be printed or saved to file, too. By default is set simple text editor **Notepad.exe**, which is standard part of all versions of MS Windows.

In Print buffer dialog are following options:

Block start
Defines start address of selected block in buffer.
Block end
Defines end address of selected block in buffer.

External editor
Defines path and name of external program, which has to be used as text viewer for selected block of buffer. By default is set simple text editor Notepad.exe, which is standard part of all versions of MS Windows. User can define any text editor for example Wordpad.exe, which is able to work with large text files. In user defined text editor user can print or save to file selected block of buffer.
The external editor path and name is saved automatically to disk.

Find dialog box

Enter the search string to **Find** to text input box and choose **<Find>** to begin the search or choose **<Cancel>** to forget it.

Direction box specifies which way you want to search, starting from the current cursor position (In edit mode). **Forward** (from the current position or start of buffer to the end of the buffer) is

the default. **Backward** searches toward the beginning. In view mode searches all buffer.

Origin specifies where the search should start.

Find & Replace dialog box

Enter the search string in the **Text to find** string input box and enter the replacement string in the **Replace with** input box.

In **Options** box you can select prompt on replace: if program finds instance you will be asked before program change it.

Origin specifies where the search should start.

Direction box specifies which way you want to search, starting from the current cursor position (In edit mode). **Forward** (from the current position or start of buffer to the end of the buffer) is the default. **Backward** searches toward the beginning. In view mode searches all buffer.

Press **<Esc>** or click **Cancel** button to close dialog window.

By pressing **Replace** button the dialog box is closed and a Question window is displayed. This window contains following choices:

Yes	replaces found item and finds next
No	finds next item without replacing current one
Replace All	replaces all found items
Abort search	aborts this command

View/Edit buffer for PLD

Ctrl+F2	erase buffer with specified blank value
Ctrl+Shift+F2	fill buffer with random data
F9	go to address...
F10	change mode view / edit
F11	switch the mode of buffer data view between 1 bit and 8 bit view. It can be also do by mouse clicking on the button to the right of View/Edit mode buffer indicator. This button indicates actual data view mode (1 bit or 8 bit), too.
Arrow keys	move cursor up, down, right and left
Home/End	jump on start / end current line
PgUp/PgDn	jump on previous / next page
Ctrl+PgUp/PgDn	jump on start / end current page
Ctrl+Home/End	jump on start / end edit area
Backspace	move cursor one position left (back)

Note: *Characters 0 and 1 immediately changes content of edit area.*

Buffer / Fill block

Selecting this command causes filling selected block of buffer by requested hex (or ASCII) string. Sets start and end block for filling and requested hex or ASCII string.

Buffer / Copy block

This command is used to copy specified block of data in current buffer on new address. Target address needn't be out from source block addresses.

Buffer / Move block

This command is used to move specified block of data in current buffer on new address. Target address needn't be out from source block addresses. Source address block (or part) will be filled by topical blank character.

Buffer / Swap block

This command swaps a high- and low- order of byte pairs in current buffer block. This block must started on even address and must have an even number of bytes. If this conditions do not fulfil, the program modifies addresses itself (start address is moved on lower even address and/or end address is moved on higher odd address).

Buffer / Erase

If this command is selected, the content of the buffer will be filled with topical blank character.

The reserved key **<Ctrl+F2>** will bring out this menu from any menu and any time.

Buffer / Fill random data

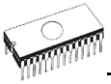
If this command is selected, the content of the buffer will be filled with random data.

The reserved key **<Shift+Ctrl+F2>** will bring out this menu from any menu and any time.

Buffer / Duplicate buffer

This command performs duplicate buffer content in range of source EPROM to range of destination EPROM. This procedure is suitable if there is used for example 27C512 EPROM to 27C256 EPROM position.

Note: *The procedure always uses buffer start address 00000h.*



Buffer / Checksum

The checksum dialog is used for calculate checksums of selected block in buffer. The checksums are calculated by next way :

Byte	sum by bytes to "word". CY flag is ignored
Word	sum by words to "word". CY flag is ignored
Byte (CY)	sum by bytes to "word". CY flag is added to result.
Word (CY)	sum by words to "word". CY flag is added to result.
CRC-CCITT	sum by bytes to "word" using RESULT=PREVIOUS + (x^16 + x^12 + x^5 +1)
CRC-XModem	sum by bytes to "word" using RESULT=PREVIOUS + (x^16 + x^15 + x^2 +1)

Column marked as **Neg.** is a negation of checksum so, that Sum + Neg. = FFFFH.
Column marked as **Suppl.** is complement of checksum so, that Sum + Suppl. = 0 (+ carry).

Dialog checksum contains following items:

From address: This is a start address of block selected for calculating checksums in buffer. Address is defined as Byte address.

To address: This is an end address of block selected for calculating checksums in buffer. Address is defined as Byte address.

Insert checksum: This is special item used for select which kind of checksum will be written into the buffer when, the **Calculate & insert** was executed.

Insert address: This is special item that specifies an address from the buffer where a result of chosen checksum will be written, when the **Calculate & insert** was executed. Address can not be specified inside the range **<From address>** to **<To address>**. Address is defined as Byte address.

Size: This item is used for setting a size of chosen checksum result, which will be written into the buffer. A size of checksum result may be 8 (byte) or 16 (word) bits long. If word size was selected, whole checksum value will be written into the buffer. In other case will be written only low byte of checksum value.

Note: *If word size was selected, a low byte of checksum value will be written on address specified in box Insert address and a high byte will be written on address incremented by one.*

Calculate: Click on the button Calculate starts calculating checksums for selected block in buffer. No writes into the buffer are executed.

Calculate & insert: Click on the button **Calculate & insert** starts calculating checksums for selected block in the buffer and writes the chosen checksum into the buffer on address specified by **Insert address**.

Options

The Options menu contains commands that let you view and change various default settings.

Options / General options

General options dialog allows user to control following options of program.

File options

File options page allows you to set file masks, auto-reload of current file and choose file format recognizing for loaded files.

File format masks is used for setting file-name masks to use as a filter for file listing **in File / Save** and **File / Load file** window for all file formats. Mask must contain one of wildcards (*, ?) at least and must be applied correctly by syntax.

Project file default extension is used for setting project files-extension used as default extension in **File / Load project** and **File / Save project** dialogs.

In group **When current file is modified by another process** can be set mode of reloading of actually loaded (current) file. There are three choices:

1. Prompt before reloading file
2. Reload automatically
3. Ignore change scanning of current file

Load file format allows to set mode of file format recognition for loading files. When automatic file format is selected, program analyses format of loading file and test file for each of supported formats that are available in program. If file format matches one of supported formats, the file is read to buffer in detected format.

Manual file format allows user to select explicitly wished file format from list of supported file formats. File may be loaded no completely or incorrectly, if file format does not match to user selected format.

Hex file options

This page contains several options for loading control by any of HEX formats.
The first option sets **erasing** buffer (with desired value) automatically before the loading by any of HEX formats.
The second option sets a **negative offset**, which is used for data addresses modification by loading from any HEX file so, that data can be written to existing buffer addresses.

Example:
A file contents data by Motorola S - format. A data block started at address FFFF0H. It is a S2 format with length of address array of 3 bytes. For all data reading you can set a value of negative offset to FFFF0H. It means, that the offset will be subtracted from current real addresses and so data will be written from buffer address 0.

Warning: *The value of negative offset is subtracted from real address and therefore a result of subtraction can be negative number. Because take care of correct setting of this value.*

The third option sets data **redirecting** from file by extended Intel HEX format to physically existing buffer addresses. It means, that the user will be prompted to enter segment, which is associated with the lowest segment in file (record type 02) and the other segments will be modified with this difference. This option isn't valid for "simple" Intel HEX format. Default set means inactive status.

Example:
The file contains two records of type 02 with addresses F000H and F800H. When we enter new segment for example 0H, data from segment F000H will be directed to segment 0H and similar data from segment F800H to segment 800H.

Default setting is no active.

Language

This page allows you to select another language for user interface such as menu, buttons, dialogs, information and messages. It also allows to select wished help file in another language. For another language support of user interface the language definition file is required.

Sound

Sound page allows user to select the sound mode of program. Program generates sounds after some activities, e.g. activities on device (programming, verifying, reading, etc.). Program generates sound also when warning or error message is displayed. User can now select sound from MS Windows system sound (required installed sound card), PC speaker or none sound.

Other

Page Other allows user to manage other program settings. In the panel Tool buttons, hint display options on toolbar buttons in main program window can be modified. In the panel Start-up directory can be selected mode of selecting directory when program starts. Default start-up directory means directory, from which program is called. Directory in which program was lastly ended means the last current directory when program was lastly ended. This directory assumes the first directory from directory history list.

Save options

Save options page allows you to select the program options saving when exiting program. Three options are available here:

Don't save options - don't save options during quitting program and don't ask for saving options.
Auto save options - save options during quitting program without asking for saving options.
Prompt for save options - program asks user for saving options before quitting program. User can select to save or not to save options.

Options / View

Use the View menu commands to display or hide different elements of program environment such as toolbars.

Options / View / Main toolbar

Choose this command to show or hide the Main toolbar.

Options / View / Additional toolbar

Choose this command to show or hide the Additional toolbar.

Options / View / Device options before device operation

Choose this command to enable/disable display of Device options before device operation is confirmed.

Options / Display errors

This option allows you set a form of errors displaying as a result of programmed data verifying. Errors can be displayed to the **screen** (max. 45 differences), **saved to VERIFY.ERR** file on the disk in current directory or it will **not displayed**. In case the displaying errors are turned off, the control program reports a warning message in INFO window only.

This setting can be saved to disk by command **Options / Save options**. Default form is set to a screen displaying.

Options / Find programmer

Selects a new type of programmer and communication parameters. This command contains following items:

Programmer - sets a new type of programmer for find. If a Search all is selected, the control program finds all supported programmers.
Establish communication - allows manual or automatic establishing communication for a new programmer.
Speed - sets speed, if a manual establishing communication is selected, which PC sends data into the programmer. Speed is expressed as a percent from a maximal speed.

The communication speed modification is important for PCs with "slow" LPT ports, which haven't sufficient driving power for a PC<->programmer cable (laptop, notebook, ...). Use this command, if you have any communication problems with connected programmer on the LPT port of your PC (e.g. control program reports a programmer absence, the communication with the programmer is unreliable, etc.).

If automatic establishing communication is selected, then control program sets a maximal communication speed.

Port - selects a LPT port, which will be scanned for a requested programmer. If All port is selected, the control program scans all LPT ports, which are available on standard addresses.

Address for special port - sets address of LPT port, if a Special port is selected.

Pressing key **<Enter>** or button **OK** initiates scanning for programmer by set parameters. There is same activity as at start the control program. The command clears a list of default devices without the current device, if the new selected programmer supports this one.

This setting is saved to disk by command **Options / Save options**.

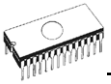
Options / Handler

In dialog **Handler** a Handler type and Handler communication parameters can be set. Handler is an external device for special control of device operations in control program. When None Handler is selected, this means default state of control program, i.e. device operations are controlled directly by user otherwise control program is in special mode, when device operations are controlled automatically with co-operation with Handler.

Dialog **Handler** contains following items:

Selected Handler select wished Handler type.
Search at port select a COM port, which will be scanned for a requested Handler.

Pressing key **<Enter>** or button **OK** initiates scanning for Handler by set parameters. If selected Handler type is **None**, no Handler scanning will be processed. Current Handler settings are saved to configuration file by command **Options / Save options** or when control program is closed.
Handler is not available for sale.



Options / Module options

This option is used for multiple socket programmers for defining MASTER socket and activity of each socket. **MASTER socket** group box allows user to set socket which is preferentially used for device reading operation. **Enable/Disable socket** checkbox array allows user to set enabling and disabling of each socket individually. Disabled sockets are ignored for any device operation.

Options / Automatic YES!

This command is used for setting **Automatic YES!** mode. In this mode you just put a device into ZIF socket and a last operation will be repeated automatically. Program automatically detects an insertion of a new device and runs last executed operation without pressing any key or button. An insertion of device into ZIF is displayed on the screen. Repeated operation executing will be cancelled by pressing key **<Esc>** during waiting for insert/remove a device to/from ZIF.

Note: *During waiting for an insertion a new device into ZIF socket, the LED BUSY on the programmer is blinking.*

This mode may be enabled or disabled by item **Automatic YES!** mode. If a new programmer is selected **Options / Find programmer**, this mode will be disabled.

In **Response time** is possible to set a time interval within must be detected device in ZIF socket to accept an insertion of a new device. Default is set standard interval. If socket adapter is used then is recommended to set an elongated interval.

In **Pins with capacitors** bar may be entered a list of a pins interconnected by capacitors (for example: if a converter, which have connected capacitor between VCC and GND, is used), which may makes problems at detecting insertion of a new device.
List of pins of device is in form:
pinA, pinB, pinC....
Example: 4,6,17

This list is erased if a new device is selected by **Device / Select default** or **Device / Select device** ...

This setting is saved to disk by command **Options / Save options**.

Options / Log file

This options associates with using of **Log window**. All reports for this window can be written into the **Log file** too. The Log file name as **REPORT.REP** and the control program creates this file in current directory.
Sets **New** caused deleting old Log file if exist and creating a new file for reports. Sets **Append** adds all reports into existing Log file. If file not exist, the new file will be created. Settings are applied only at program start.

This setting can be saved to disk by command **Options / Save options**. Default form is set to a using Log window without Log file i.e. all reports will be displayed to a Log window only.

Options / Protected mode

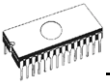
Protected mode is special mode of program. When program is in Protected mode, there are disabled program operation and commands that can modify buffer or device settings. Protected mode is used for prevent operator from modify buffer or device settings due to insignificance. Protected mode is suitable for the programming of a large amount of the same type of devices.

There are two ways how to switch program to Protected mode:

1. by using menu command **Options / Protected mode**. This command displays password dialog. User have to enter password twice to confirm the password is correct. After password confirmation program switches to Protected mode. The entered password is then used to switch off Protected mode.
2. by reading project, which was previously saved in Protected mode.

To switch program from Protected mode to normal mode use the menu command **Options / Normal mode**. The "Password required" dialog appears. User have to enter the same password as the password entered during switch to Protected mode.

Other way to cancel Protected mode of program is closing of program, because program Protected mode is active until program is closed. The next program start will be to Normal (standard) mode (the only exception is case of project loaded by command line parameter name of project and the project was saved in Protected mode).



Options / Save options

This command saves all settings that are currently supported for saving, even if auto-save is turned off. Following options are saved: options under the Options menu, ten last selected devices, file history, main program window position and size.

Diagnostics

This command includes selftest for programmers and IC test.

Diagnostics / Selftest

Command executes a selftest of current programmer without diagnostic POD. We recommend execute also **Diagnostics / Selftest plus** of programmer.

Diagnostics / Selftest plus

Command executes a selftest of current programmer using diagnostic POD, which is included in standard delivery of programmer. We recommend run this test as often as possible, e.g. once per month.

Diagnostics / IC test

This command activates a test section for ICs separated by compatibility to any libraries (on distribution diskette). First select an appropriate library, wished device and then a mode for test vectors run (**Loop**, **Single step**). Control sequence and test results are displayed to Log window.

Diagnostics / Create diagnostic report

Command Create Diagnostic report is used for writing more particular diagnostic information to **Log window** and consequently copy **Log window** content to clipboard. The Log window content can be placed from clipboard to any text editor. Diagnostic report is useful when error occurs in control program or programmer and kind of the error is, that user can not resolve it oneself and he must contact programmer manufacturer. In this case when customer send message to manufacturer about his problem it is good to send also diagnostic report. Diagnostic report can help manufacturer to localise the reason of error and resolve it sooner.

Help

Pressing the **<F1>** key accesses the Help. When you selecting menu item and press **<F1>**, you access context-sensitive help. If PG4U / PG4UW is executing an operation with the programmer **<F1>** generates no response.

The following Help items are highlighted:
- words describing the keys referred to by the current Help
- all other significant words
- current cross-references; click on this cross-reference to obtain further information.

> *Since the Help system is continuously updated together with the control program, it may contain information not included in this manual.*

Detailed information on individual menu commands can be found in the integrated on-line Help.

Note: *Information provided in this manual is intended to be accurate at the moment of release, but we continuously improve all our products. Please consult manual on www.elnec.com.*

Help / Supported devices

This command displays list of all devices supported by at least one type of all supported programmers. It is useful especially when user wants to find any device supported by at least one type of programmers.
Prefix "g_" before name of device means the device is supported by multi-socket programmer.

Help / Supported programmers

This command displays information about programmers, where supported this program.

Help / Device list (current programmer)

This command makes a list of all devices supported by current programmer and saves its to **?????DEV.txt** text file and **?????DEV.htm** HTML file in the directory where control program is run from. Marks ????? are replaced by abbreviated name of current programmer, the device list is generated for.

Help / Device list (cross reference)

This command makes cross reference list of all devices supported by all programmers available on market and supported by this control program. The resulting list is in HTML format and consists of following files:

- one main HTML file **TOP_DEV.htm** with supported device manufacturers listed
- partial HTML files with list of supported devices for each device manufacturer

Main HTML file is placed to directory where this control program for programmers is located.

Partial HTML files are placed to subdirectory **DEV_HTML** placed to the directory where control program for programmers is located.

About

When you choose the Info command from the menu, a window appears, showing copyright and version information.

.

Common notes

Software

PG4U/PG4UW is common control program for all of the ELNEC programmers. Thus, during work with him its possible to find some items, those refer not to current selected programmer.

There is possible to attach to selected file "device associated file" too. This file may be automatically loaded into the buffer, if item **Device options/Associated file/Automatic load** is set to Yes. File can be associated manually or accepting last loaded file into the buffer. For this purpose is used item **Device options/Associated file/File name**.

There exist feature of control program, named **Automatic YES!**. In concrete, it mean there isn't necessary to press "YES!" labeled button to repeat last activity, because auto-sensing feature of programmer detect insertion of new chip into ZIF socked and start last executed automatically. This feature can be set in menu Options. If you use package converter with come pins coupled by capacitor, enter list of these pins. **Automatic YES!** feature is available for JetProg, LabProg+, and SmartProg programmer only.

Some special devices (e.g. Philips Coolrunner family) require external DAT files, that isn't present in standard PG4U/PG4UW SW delivery on CD. If you need to program these devices, look at **www.elnec.com**, section Download.

Hardware

Due a large variety of parallel port types, a case may occur when the programmer cannot "get concerted" with the PC. This problem may be shown as none communication between the PC and the programmer, or by unreliable communication. If this behavior occur, try to connect your programmer to some other PCs or other parallel ports near you.

If you find none solution, please document the situation, i.e., provide us an accurate description of your PC configuration, including some other circumstances bearing on the problem in question, and advise the manufacturer of your problem. Don't forget please enter of PC type, manufacturer, speed, operation system, resident programs; your parallel port I/O manufacturer and type. Use please **Device problem report** form for this purpose (see **Appendix A**).

ISP (In-System Programming)

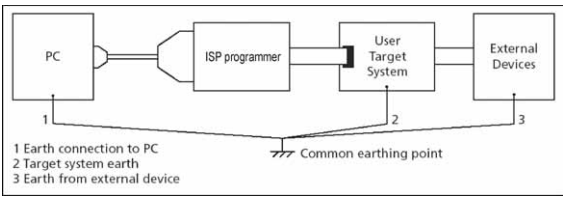
Definition

In-system programming allows programming and reprogramming of device positioned inside the end system. Using a simple interface, the ISP programmer communicates serially with the device, reprogramming nonvolatile memories on the chip. In-system programming eliminates the physical removal of chips from the system. This will save time, and money, both during development in the lab, and when updating the software or parameters in the field.
Target device is the device (microcontroller, PLD, etc…), which is to be in-system programmed.
Target system is the physical Printed Circuit Board (PCB), which contains the device to be in-system programmed.
ISP programmer is programmer, which has in-system programming capability (for example SmartProg, T51prog, PIKprog+…).

General rules for in-system programming

We recommended respect following rules to avoid damage PC, ISP programmer, and target device or target system:

- Ensure common earth point for target system, ISP programmer and PC.
- For laptop or other PC that is not connected to common earth point: make hard - wired connection from laptop to common earth point (for example use LPT or COM port D – connector).
- Any devices connected to target system must be connected to common earth point too.



Direction of connect ELNEC ISP programmer to target system:

During in-system programming you connect two electrical devices – ISP programmer and target system. Unqualified connection can damage these devices.

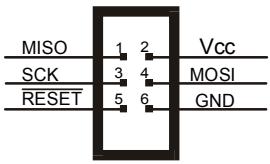
Note: *When you don't keep below directions and you damage programmer during in-system programming, it is damage of programmer by unqualified manipulation and is out of warranty.*

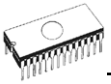
1. Turn off both devices – ISP programmer and target device.
2. Assign same GND potential for all devices, e.g. connect GND of all devices by wire.
3. Insert one connector of ISP cable to ISP programmer, turn on programmer and control program.
4. In control program select target device and operation options.
5. Start action on target device (read, program).
6. After direction of control program, connect other ISP cable connector to target system and turn on it.
7. After direction of control program, disconnect other ISP cable connector from target system and turn off it.
8. If you need another action on target device, you continue with step 5.

The recommendation for design of target system with ISP programmed device

The target system must be designed to allow all signals, which are use for In-system programming to be directly connected to ISP programmer via ISP connector. If target system use these signals for other function, is necessary isolated these signals. Target system mustn't affect these signals during In-system programming.
For in-system programmable devices manufacturers publish application notes. Design of ELNEC programmers together with respect of these application notes allow proper In-system programming. Condition is exactly respect these application notes. Applications notes, which ELNEC use in ISP programmers are published in **www.elnec.com**, section Application notes.

Example of application note
Microcontrollers Atmel AVR and AT89Sxxx series
This application note is used in T51prog. This interface corresponds with Atmel application note AVR910: In-System Programming. This application note describes the recommended ISP interface connector layout in target system (top view).





Description of required pins for in-system programming by T51prog.

Pin	Name	Comment
SCK	Serial Clock	Programming clock, generated by the In-System programmer (master).
MOSI	Master Out – Slave In	Communication line from In-System programmer (master) to target MCU being programmed (slave).
MISO	Master In – Slave Out	Communication line from target MCU (slave) to In-System programmer (master).
GND	Common Ground	The two systems must share the same common ground.
RESET	Target MCU Reset	To enable In-System programming, the target MCU Reset must be kept active. To simplify this, the In-System programmer should control the target MCU Reset
Vcc	Target Power	To allow simple programming of targets operating at any voltage, the In-System programmer can draw power from the target. Alternatively, the target can have power supplied through the In-System programming connector for the duration of the programming cycle

PICmicro® microcontrollers

This application note is used in PIKprog+. This interface corresponds with Microchip application notes TB013, TB017, TB016: How to Implement ICSP™ Using PIC16CXXX OTP (PIC12C5XX OTP)(PIC16F8X Flash) MCUs. These application notes describes requirement for target system with In-system programming device and ISP programmer.

Following signals are use for In-system programming of PICmicro® microcontrollers.

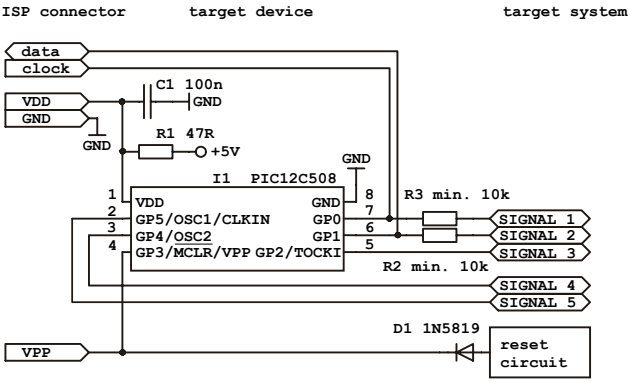
MCLR\ / VPP	reset / switch to programming mode
RB6 (GP1)	clock
RB7 (GP0)	data input / output
VDD	power supply
GND	ground

When PICmicro® device is programmed, pin MCLR\ / VPP is driven to approximately 12 V. Therefore, the target system must be isolated from this voltage provided by programmer.

RB6 and RB7 signals are used by the PICmicro® for In-system programming, therefore target system mustn't affect these signals during In-system programming to avoid programming errors.

Marginal verify is used after programming. Programmer must verify the program memory contents at both minimal and maximal power supply, therefore VDD pin of PICmicro® must be isolated from rest of target system during programming.

ELNEC's recommended circuit:



Other

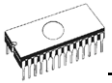
Attention to multitasking OS's (WIN 3.11/95/98/Me/NT/2000/XP). There is needful for regular running of control program for any ELNEC programmer that printer port, on which is programmer connected, must be reserved for this programmer only. Otherwise, any other program must not simultaneously to use (or any way to modify) this printer port.

PG4U/PG4UW SW can handle all modes of LPT port (full IEEE 1284 support), thus you don't need to configure LPT port for connection of ELNEC programmers.

WIN98 have bug (or wittingly) in the MSDOS.SYS file. Initial setting of variable DoubleBuffer is 1, therefore DOS applications run slowly. Write please DoubleBuffer=0.

In case of WIN software, please don't move any window during BUSY LED is on - watching circuit can be activate to switch the programmer in safe status as in case communication PC-programmer error.

Troubleshooting and warranty



Troubleshooting

We really want you to enjoy our product. Nevertheless, problems can occur. In such cases please follow the instructions below.

- It might be your mistake in properly operating the programmer or its control program PG4U/PG4UW.
  - Please read carefully all the enclosed documentation again. Probably you will find the needed answer right away.
  - Try to install programmer and PG4U/PG4UW on another computer. If your system works normally on the other computer you might have a problem with the first one PC. Compare differences between both computers.
  - Ask your in-house guru (every office has one!).
  - Ask the person who already installed programmer.

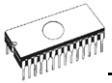
- If the problem persists, please call the local dealer, from whom you purchased the programmer, or call ELNEC direct. Most problems can be solved by phone, e-mail or fax. If you want to contact us by:
  - **Mail/fax** - Copy the "**DEVICE PROBLEM REPORT**" form and fill it in following the instructions at the end of the form. Write everything down that you consider to be relevant about the programmer, software and the target device. Send the completed form by mail or fax to ELNEC (fax number in the control program, menu **Help / About**) or to your local dealer. If you send the form by fax please use black ink, a good pen and large letters!
  - **E-mail** - Use "**DEVICE PROBLEM REPORT**" form on the CD or from our Internet site and fill it in following the instructions at the end of the form. Use standard ASCII editor. Write everything down that you consider to be relevant about the programmer, software and the target device. Send the completed form by e-mail to your local dealer or to ELNEC (**elnec@elnec.com**).
  - **Phone** - Copy "**DEVICE PROBLEM REPORT**" form and fill it in following the instructions at the end of the form. Write everything down that you consider to be relevant about the programmer, software and the target device. Send the completed form by mail or fax to ELNEC (fax number in the control program, menu **Help / About**) or to your local dealer. If you send the form by fax please use black ink, a good pen and large letters easily to read. Then call your local dealer or ELNEC's customer support center (phone number in PG4U, submenu **Info**, in PG4UW **Help / About**). Please keep your manual, the programmer and the completed "**DEVICE PROBLEM REPORT**" form (just faxed) available, so that you can respond quickly to our questions.

- If your programmer is diagnosed as defective, consult your local dealer or ELNEC about the pertinent repair center in your country. Please carefully include the following items in the package:
  - defective product
  - completed "**DEVICE PROBLEM REPORT**" form
  - photocopy of a dated proof of purchase

Without all these items we cannot admit your programmer to repair.

Note:
*You may find the "**DEVICE PROBLEM REPORT**" form:*

- *in **Appendix A** of this manual*
- *on the enclosed CD (file **DEVPROR.TXT**)*
- *at our Internet site (**www.elnec.com**), section 'Notes'.*
- *if the period between purchasing the programmer and the defective incident is greater than 6 months please also include the "**Extended warranty**" document with your own registration number. For more information please read the chapter "**Warranty terms**".*



If you have an unsupported target device

If you need to operate on a target device not supported by the control program for programmer, please do not despair and follow the next steps:

- Look in the device list of the latest version of the control program on our Internet site (section Download, file corresponded to your programmer). Your new target device might already be included in this version! If yes, download the file PG4UARC.EXE (for DOS) or PG4UWARC.EXE (for WIN) and install the new version of the control program.
- Contact ELNEC direct, filling up a "**Device Problem Report**" form following the instructions at the end of this form. We may need detailed data sheets of your target device and, if possible, samples. The samples will be returned to you after we include your target device in a new version of PG4U/PG4UW.

Note:

See also AlgOR service in Appendix C in this manual.
*You may find the "**Device Problem Report**" form:*

- *in **Appendix A** of this manual*
- *on the enclosed CD (file **DEVPROR.TXT**)*
- *at our Internet site (**www.elnec.com**), section 'Notes'.*

Warranty terms

The manufacturer, ELNEC s.r.o. Presov, Slovakia, gives a guarantee on failure-free operating of the programmer and all its parts, materials and workmanship for **one-year** (T51prog, 51&AVRprog, PIKprog+, PIKprog and SEEprog) or **three-year** (JetProg, LabProg+, SmartProg, PREPROM-02aLV) from the date of purchase. This warranty is limited to 25,000-cycles on DIL ZIF socket or 10,000-cycles on PLCC ZIF sockets). If the product is diagnosed as defective, ELNEC s.r.o. or the authorized repair center will repair or replace defective parts at no charge. Parts used for replacement and/or whole programmer are warranted only for the reminder of the original warranty period.

For repair within the warranty period, the customer must prove the date of purchase. Extended warranty is available only for customers who have the "**Extended warranty**" document with their own registration number from ELNEC (please read note at the end of this chapter). All other programmer owners only carry a standard 6 months warranty.

The warranty does not apply to products that are of wear and tear or mechanically damaged. Equally, the warranty does not apply to products opened and/or repaired and/or altered by personnel not authorized by ELNEC, or to products that have been misused, abused, accidentated or that were improperly installed.

For unwarrantable repairs you will be billed according to the costs of replacement materials, service time and freight. ELNEC or its distributors will determine whether the defective product should be repaired or replaced and judge whether or not the warranty applies.

Note:

- *Every customer obtains an "**Extended warranty**" document and an own registration number if he/she sends to ELNEC the filled out **registration** form within 4 weeks from the purchase of the programmer. The Registration form is part of the standard programmer package. If this form is missing, please copy it from **Appendix D** of this manual.*
- *Please also see Troubleshooting section.*

Manufacturer:
✉: **ELNEC s. r. o.**, Post 5, P. O. Box 22, SK - 08005 Presov, Slovakia
☎: +42151/77 34 328, 77 31 007, fax 77 32 797
www.elnec.com, elnec@elnec.com

Appendix

Appendix A - Device Problem Report form

Please make a copy of this page to A4 or use the A4 form enclosed in standard delivery package. This form can also be found on the enclosed CD disk and on our Internet site.

DEVICE PROBLEM REPORT

Subject(title of problem):_____ Date:_____

Customer

Customer, name:_____ Distributor, name:_____
Address:_____ Date of purchasing:_____
Contact person and e-mail:_____ Date of sending registration card:_____

Information about product.

Programmer (type/modification):_____ Mains supply voltage:_____V (in case 51&AVR prog and PIK prog)
Serial number:_____ Version of control program PG4U/W:_____
Configuration (modules, converters):_____
Power supply unit: From delivery Other (output V and A): _____

Information about PC, to which is the programmer is attached.

Manufacturer/Type:_____ Desktop Notebook
Processor, speed:_____ LPT port location: motherboard ISA card PCI card
Operating system and version:_____ LPT port type: standard ECP/EPP 1284
Memory/free memory:_____ LPT port setting: SPP BIDIR EPP ECP

Information about device with which you have the problem.

Device type (full name, prefix/suffix including):_____ Package type: plastic ceramic ceramic/windowed
Vendor/logo:_____ All designation on the top_____
Package (DIL40, PLCC44, SOIC20, ...):_____ and on the bottom side of device_____

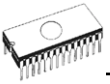
Precedence rating: in __ days in __ weeks in __ months
How often you work with this devices: still Y/N sometimes Y/N one-shot Y/N
Number of programmed device: approx. ___ pcs per year.
Samples are available? Yes (I'm sending it/attached) Yes No

Further questions.

- Did you have installed latest version of control program? Yes No
- Did you know thoroughly the features and correct behavior of programmer and programmed device? Yes No
- Is the socket of programmer or adapter free from dust and isn't out of life? Yes No
- Is the device with problem new or used? New Used
- Is the error reported for all of the tested devices? Yes No I have only one device
- Is the error reported for devices with other date code? Yes No I have only one batch
- During which procedure is an error reported? Read Program ID_check Insertion test
- Is the programmer successful in case of other types of devices? Yes No
- Does the error occur always or randomly? Always Randomly
- Does programmer work well with other PCs? Yes No Not tested
- Is the error reported in both DOS and WIN version of SW? Yes No
- What is the results of programmer selftest (if available)? OK Error

Please list the step-by-step description of all activities that invokes a problem. Please make your problem description as specific as possible - you can increase speed and chance to resolve a problem. Please mention any step that is known to cause the problem or any step that may prevent the problem. Please copy all error reports too - full content of LOG window is preferable. We recommended use command Diagnostics/ Create diagnostic report. Your comments and descriptions of expectations are welcomed. It's best, if you can send us the actual device with which the problem occurs. Use a separate sheet if necessary.

_____
_____
_____
_____
_____
_____



Note:

- *if you haven't installed the latest version of control program, you can get it from www.elnec.com page (Download section). It is very important to have latest version of software, because:*
  a) *it is possible the problem you have is already solved by software update*
  b) *we don't save older version of software. If ask you to "please perform next steps ...", your version of software may not behave in the same way as the latest one as used by us.*
- *in the case of sending samples, please attach to the package this declaration for customs: "Free sample(s), not for commercial sale.*

Value for customs purposes only: $10US"

Appendix B - Keep-Current Service

Keep-Current is a subscription service, with which your ELNEC programmer and documentation is kept up-to-date with the latest device support list and with the last version of the control program and its new features.

ELNEC ships off the latest version of the programmer software and updated user documentation (Keep-Current package). A Keep-Current service is your task-free guarantee that you are exploiting your ELNEC programmers and doing high quality programming at maximum and with minimal costs.

Keep-Current package can be mailed to you in two ways:

- by standard post
- by e-mail.

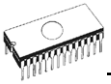
According to your needs you may choose three levels of Keep-Current service:

- **QUICK** ☞ Keep-Current package is mailed to you immediately after each relevant change of software or documentation (approximately monthly). This service carries a fee (please consult price list) and it's available only by e-mail.
- **STANDARD** ☞ Keep-Current package is mailed to you quarterly. This service again carries a cost, (please consult price list) and is available by post or by e-mail.
- **1/Y** ☞ Keep-Current package is mailed to you once per year. This service is available only by e-mail and is free - but a registration is required.

Keep-Current package (file KC_ARC.EXE) contains:

- latest version of the control program (file PG4UARC.EXE or PG4UWARC.EXE)
- latest device list
- description of added/removed device and/or modified algorithms
- description of added and/or modified features of the control program
- latest information about ELNEC's products

Prices of Keep-Current service (P&P included)		
level	*e-mail*	*per post*
QUICK	**30USD**	**not offered**
STANDARD	**15USD**	**35USD**
1/Y	**free**	**postage / packing cost**



Why is it important to use the latest version of the control program?

- Semiconductor manufacturers continuously introduce new devices with new package types, manufactured by new technologies in order to support the need for flexibility, quality and speed in product design and manufacturing. To keep pace and to keep you up-to-date, we usually implement more than 500 new devices into the control program within a year.
- Furthermore, a typical programmable device undergoes several changes during its lifetime in an effort to maintain or to improve its technical characteristics and process yields. These changes often impact with the programming algorithms, which need to be upgraded (the programming algorithm is a set of instructions that tells the programmer how to program data into a particular target device). Using the newest algorithms in the programming process is the key to obtaining high quality results. In many cases, while the older algorithm will still program the device, they may not provide the level of data retention that would be possible with an optimal algorithm. Failure to not use the most current algorithm can decrease your programming yields (more improper programmed target devices), and may often increase programming times, or even affect the long term reliability of the programmed device.
- At least, we are making mistakes too ... .

Our commitment is to implement support for these new or modified parts before or as soon as possible after their release, so that you can be sure that you are using latest and/or optimal programming algorithms that were created for this new device.

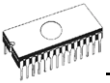
Appendix C - AlgOR service

(Algorithms On Request)
AlgOR is a free service, by which we respond, as flexible as possible on the customer's request to implement programming support for new devices. This service may be used also for requesting new features of the control program.

AlgOR process is simple. The user sends to ELNEC a request for additional support for XXX device to the control program (we may ask for up-to-date data sheets and samples, if needed). After completion, the user will obtain a new version of the control program with requested features. We will, of course, also return the borrowed samples. If we cannot satisfy your requirements (too expensive, algorithms not available, additionally module needed), we will promptly contact to you and propose an appropriate solution.

Note:

- *Please use "**AlgOR** (Algorithms On Request)" form and send it direct to ELNEC.*
- *AlgOR service is **free** of charge. Therefore we do not accept any claims regarding this service. ELNEC s.r.o. reserves the right to set the dispatching priority on the particular tasks according to its own judgment.*



Use this form please, if you request to add unsupported device into control program or you request to add/change some feature of control program. Fill-out this form completely and send it by e-mail, fax or snail-mail directly to ELNEC. Incomplete form mean lowest level of interest from customer's side. Due absence of sample may be works on the support delayed or stopped.

Please make a copy of this page to A4. This form can also be found on the enclosed CD disk and on our Internet site.

AlgOR (Algorithms On Request) form

Subject (title of problem): _____
Date: _____

Customer, name: _____
Address: _____
_____
Contact person and E-mail: _____
Distributor, name: _____
Date of purchasing: _____
Date of sending registration card: _____

Programmer (type/modification): _____
Serial number: _____
Control program and version: _____

Information about device, you want to be supported
Device type (full name): _____
Vendor/logo: _____
Package (DIL40, PLCC44,...): _____

Precedence rating: *in __ days* *in __ weeks* *in __ months*
Device to be programmed: *still Y/N* *sometimes Y/N* *one-shot Y/N*
Number of programmed device: *approx. ___ pcs per year.*
Samples are available? *Yes* *Yes (I'm sending it/attached)* *No*

Notes to request. *Description of requested change in control program.*
Enter please feature you want to the program will have.

_____
_____
_____
_____
_____

Notes.

- *look please at latest list of supported devices before you send this request to us.*
- *in case of sending samples, attach please to package next declaration for customs: "Free sample(s), not for commercial sale.*

Value for customs purposes only: $10US"

Appendix D - registration card

If the registration card is missing from your standard programmer delivery package please use a copy of the form below and send it to ELNEC. We remind you that without the "Extended warranty" document only the standard 6-month warranty is granted.

✂----------

Programmer: **LabProg+**
Serial number: _____
Name: _____
Company: _____
Department: _____
Address: _____
Post Code, City: _____
Country: _____
Phone/Fax/e-mail: _____
Date of purchase: _____
Purchase from: _____

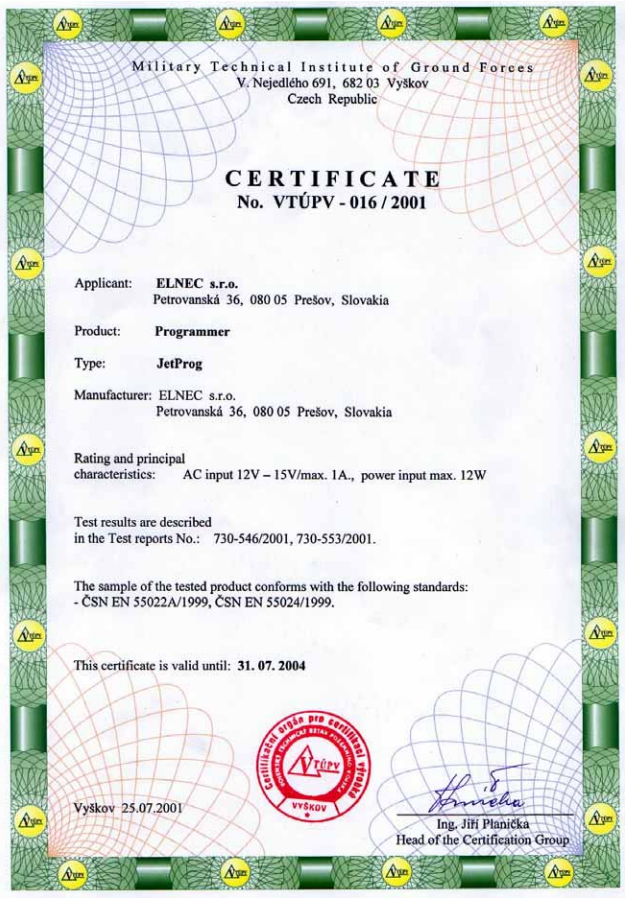
Complete, please:

- what type of computer are you using:
  ❒AT/386 ❒486 ❒Pentium I-II ❒Pentium III-4
- operating system:
  ❒MS-DOS ❒WINDOWS 3.x ❒WINDOWS 95/98
  ❒WINDOWS NT/2000/XP
- how did you first hear about ELNEC ?
  ❒advert. ❒dealer ❒other_____
- why did you choose this product?
  ❒price ❒quality ❒recommendation
  ❒features (please specify) _____
- comments:_____

_____
_____
_____
_____

Appendix E - CE certificates

E/1 - CE certificate JetProg



Military Technical Institute of Ground Forces
V. Nejedlého 691, 682 03 Vyškov
Czech Republic

CERTIFICATE
No. VTÚPV - 016 / 2001

Applicant: **ELNEC s.r.o.**
Petrovanská 36, 080 05 Prešov, Slovakia

Product: **Programmer**

Type: **JetProg**

Manufacturer: ELNEC s.r.o.
Petrovanská 36, 080 05 Prešov, Slovakia

Rating and principal characteristics: AC input 12V – 15V/max. 1A., power input max. 12W

Test results are described in the Test reports No.: 730-546/2001, 730-553/2001.

The sample of the tested product conforms with the following standards:
- ČSN EN 55022A/1999, ČSN EN 55024/1999.

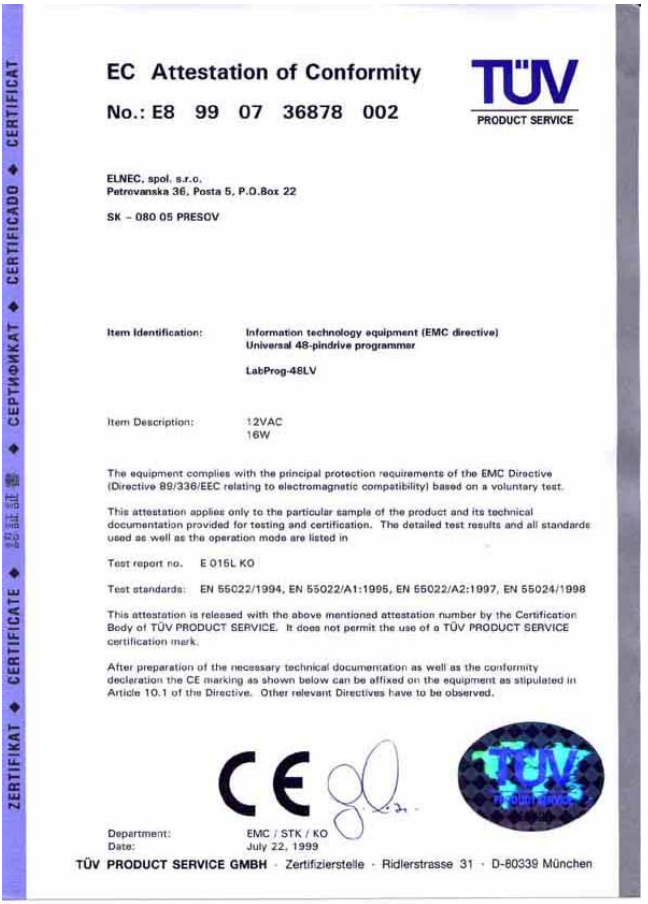
This certificate is valid until: **31. 07. 2004**

Vyškov 25.07.2001

Ing. Jiří Planička
Head of the Certification Group

E/2 - CE certificate LabProg+

LabProg+ is a new name of LabProg-48LV.



ZERTIFIKAT ◆ CERTIFICATE ◆ 認証書 ◆ СЕРТИФИКАТ ◆ CERTIFICADO ◆ CERTIFICAT

EC Attestation of Conformity

No.: E8 99 07 36878 002

TÜV PRODUCT SERVICE

ELNEC, spol. s.r.o.
Petrovanska 36, Posta 5, P.O.Box 22

SK – 080 05 PRESOV

Item Identification: **Information technology equipment (EMC directive)**
Universal 48-pindrive programmer

LabProg-48LV

Item Description: 12VAC
16W

The equipment complies with the principal protection requirements of the EMC Directive (Directive 89/336/EEC relating to electromagnetic compatibility) based on a voluntary test.

This attestation applies only to the particular sample of the product and its technical documentation provided for testing and certification. The detailed test results and all standards used as well as the operation mode are listed in

Test report no. E 015L KO

Test standards: EN 55022/1994, EN 55022/A1:1995, EN 55022/A2:1997, EN 55024/1998

This attestation is released with the above mentioned attestation number by the Certification Body of TÜV PRODUCT SERVICE. It does not permit the use of a TÜV PRODUCT SERVICE certification mark.

After preparation of the necessary technical documentation as well as the conformity decleration the CE marking as shown below can be affixed on the equipment as stipulated in Article 10.1 of the Directive. Other relevant Directives have to be observed.

CE

Department: EMC / STK / KO
Date: July 22, 1999

TÜV PRODUCT SERVICE GMBH · Zertifizierstelle · Ridlerstrasse 31 · D-80339 München

E/3 - CE certificate SmartProg

Military Technical Institute of the Ground Forces
V. Nejedlého 691, 682 03 Vyškov, Czech Republic

CERTIFICATE
Nº VTÚPV - 015 / 2002

Applicant: **ELNEC s.r.o.**
Petrovanská 36, 080 05 Prešov, Slovakia

Product: **Programmer**

Type: **SmartProg**

Manufacturer: ELNEC s.r.o.
Petrovanská 36, 080 05 Prešov, Slovakia

Rating and principal characteristics: DC input 15V / 500 mA; power input max. 6W

Test results are described in the Test reports No.: 730-384/2002, 730-389/2002.

The sample of the tested product conforms with the following standards:

- ČSN EN 55022:1999+A1 (which is identical with EN 55022:1998+A1);
- ČSN EN 55024:1999 (which is identical with EN 55024:1998).

This certificate is valid until: **31. 07. 2005**

Vyškov 18.07.2002

Ing. Jiří Planička
Head of the Certification Group

E/4 - CE certificate PREPROM-02aLV

ZERTIFIKAT ◆ CERTIFICATE ◆ 認証書 ◆ СЕРТИФИКАТ ◆ CERTIFICADO ◆ CERTIFICAT

EC Attestation of Conformity

No.: E8 99 07 36878 001

TÜV PRODUCT SERVICE

ELNEC, spol. s.r.o.
Petrovanska 36, Posta 5, P.O.Box 22

SK – 080 05 PRESOV

Item Identification: **Information technology equipment (EMC directive)**
Universal EPROM programmer

PREPROM-02aLV

Item Description: 12V DC
6W

The equipment complies with the principal protection requirements of the EMC Directive (Directive 89/336/EEC relating to electromagnetic compatibility) based on a voluntary test.

This attestation applies only to the particular sample of the product and its technical documentation provided for testing and certification. The detailed test results and all standards used as well as the operation mode are listed in

Test report no. E 016L KO

Test standards: EN 55022/1994, EN 55022/A1:1995, EN 55022/A2:1997, EN 55024/1998

This attestation is released with the above mentioned attestation number by the Certification Body of TÜV PRODUCT SERVICE. It does not permit the use of a TÜV PRODUCT SERVICE certification mark.

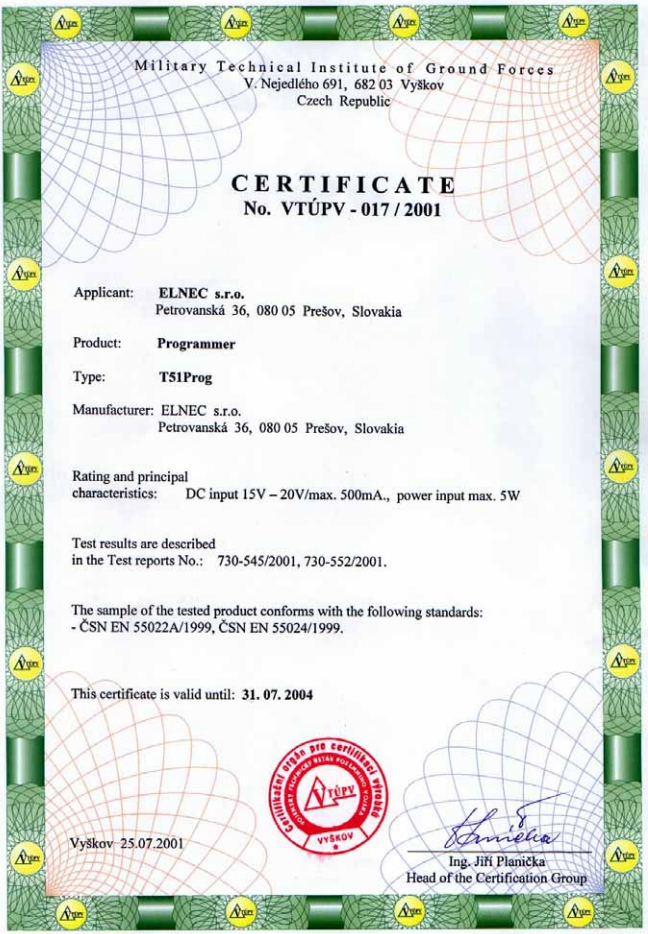
After preparation of the necessary technical documentation as well as the conformity declaration the CE marking as shown below can be affixed on the equipment as stipulated in Article 10.1 of the Directive. Other relevant Directives have to be observed.

CE

Department: EMC / STK / KO
Date: July 22, 1999

TÜV PRODUCT SERVICE GMBH · Zertifizierstelle · Ridlerstrasse 31 · D-80339 München

E/5 - CE certificate T51prog



Military Technical Institute of Ground Forces
V. Nejedlého 691, 682 03 Vyškov
Czech Republic

CERTIFICATE
No. VTÚPV - 017 / 2001

Applicant: **ELNEC s.r.o.**
Petrovanská 36, 080 05 Prešov, Slovakia

Product: **Programmer**

Type: **T51Prog**

Manufacturer: ELNEC s.r.o.
Petrovanská 36, 080 05 Prešov, Slovakia

Rating and principal characteristics: DC input 15V – 20V/max. 500mA., power input max. 5W

Test results are described in the Test reports No.: 730-545/2001, 730-552/2001.

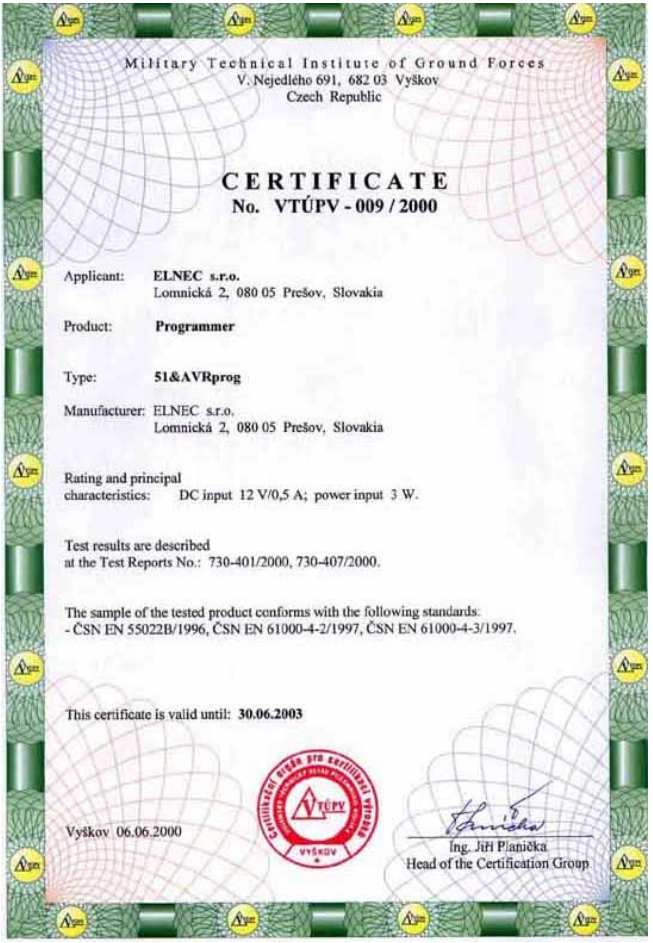
The sample of the tested product conforms with the following standards:
- ČSN EN 55022A/1999, ČSN EN 55024/1999.

This certificate is valid until: **31. 07. 2004**

Vyškov 25.07.2001

Ing. Jiří Planička
Head of the Certification Group

E/6 - CE certificate 51&AVRprog



Military Technical Institute of Ground Forces
V. Nejedlého 691, 682 03 Vyškov
Czech Republic

CERTIFICATE
No. VTÚPV - 009 / 2000

Applicant: **ELNEC s.r.o.**
Lomnická 2, 080 05 Prešov, Slovakia

Product: **Programmer**

Type: **51&AVRprog**

Manufacturer: ELNEC s.r.o.
Lomnická 2, 080 05 Prešov, Slovakia

Rating and principal
characteristics: DC input 12 V/0,5 A; power input 3 W.

Test results are described
at the Test Reports No.: 730-401/2000, 730-407/2000.

The sample of the tested product conforms with the following standards:
- ČSN EN 55022B/1996, ČSN EN 61000-4-2/1997, ČSN EN 61000-4-3/1997.

This certificate is valid until: **30.06.2003**

Vyškov 06.06.2000

Ing. Jiří Planička
Head of the Certification Group

E/7 - CE certificate PIKprog+

Military Technical Institute of the Ground Forces
V. Nejedlého 691, 682 03 Vyškov, Czech Republic

CERTIFICATE
Nº VTÚPV - 016 / 2002

Applicant: **ELNEC s.r.o.**
Petrovanská 36, 080 05 Prešov, Slovakia

Product: **Programmer**

Type: **PIKprog +**

Manufacturer: ELNEC s.r.o.
Petrovanská 36, 080 05 Prešov, Slovakia

Rating and principal characteristics: DC input 12V / 500 mA; power input max. 5W

Test results are described in the Test reports No.: 730-385/2002, 730-388/2002.

The sample of the tested product conforms with the following standards:

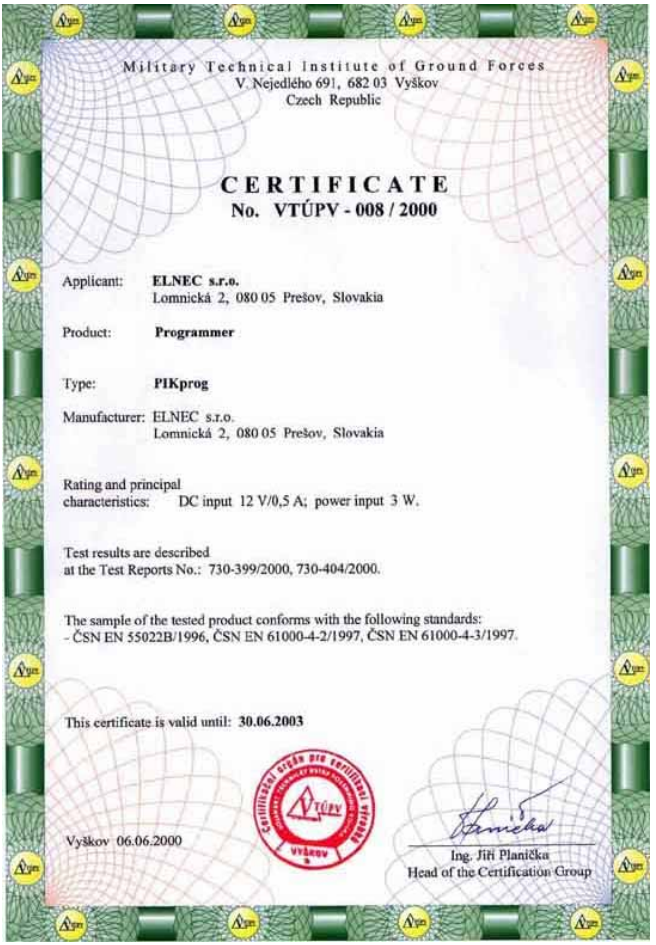
- ČSN EN 55022:1999+A1 (which is identical with EN 55022:1998+A1);
- ČSN EN 55024:1999 (which is identical with EN 55024:1998).

This certificate is valid until: **31. 07. 2005**

Vyškov 18.07.2002

Ing. Jiří Planička
Head of the Certifieation Group

E/8 - CE certificate PIKprog



Military Technical Institute of Ground Forces
V. Nejedlého 691, 682 03 Vyškov
Czech Republic

CERTIFICATE
No. VTÚPV - 008 / 2000

Applicant: **ELNEC s.r.o.**
Lomnická 2, 080 05 Prešov, Slovakia

Product: **Programmer**

Type: **PIKprog**

Manufacturer: ELNEC s.r.o.
Lomnická 2, 080 05 Prešov, Slovakia

Rating and principal characteristics: DC input 12 V/0,5 A; power input 3 W.

Test results are described at the Test Reports No.: 730-399/2000, 730-404/2000.

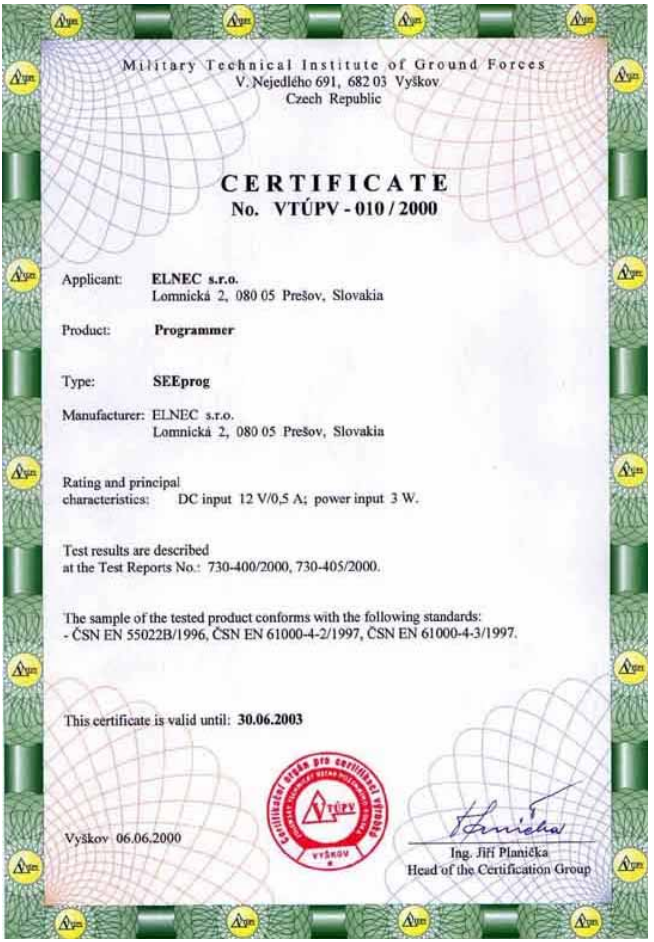
The sample of the tested product conforms with the following standards:
- ČSN EN 55022B/1996, ČSN EN 61000-4-2/1997, ČSN EN 61000-4-3/1997.

This certificate is valid until: **30.06.2003**

Vyškov 06.06.2000

Ing. Jiří Planička
Head of the Certification Group

E/9 - CE certificate SEEprog



Military Technical Institute of Ground Forces
V. Nejedlého 691, 682 03 Vyškov
Czech Republic

CERTIFICATE
No. VTÚPV - 010 / 2000

Applicant: **ELNEC s.r.o.**
Lomnická 2, 080 05 Prešov, Slovakia

Product: **Programmer**

Type: **SEEprog**

Manufacturer: ELNEC s.r.o.
Lomnická 2, 080 05 Prešov, Slovakia

Rating and principal characteristics: DC input 12 V/0,5 A; power input 3 W.

Test results are described at the Test Reports No.: 730-400/2000, 730-405/2000.

The sample of the tested product conforms with the following standards:
- ČSN EN 55022B/1996, ČSN EN 61000-4-2/1997, ČSN EN 61000-4-3/1997.

This certificate is valid until: **30.06.2003**

Vyškov 06.06.2000

Ing. Jiří Planička
Head of the Certification Group